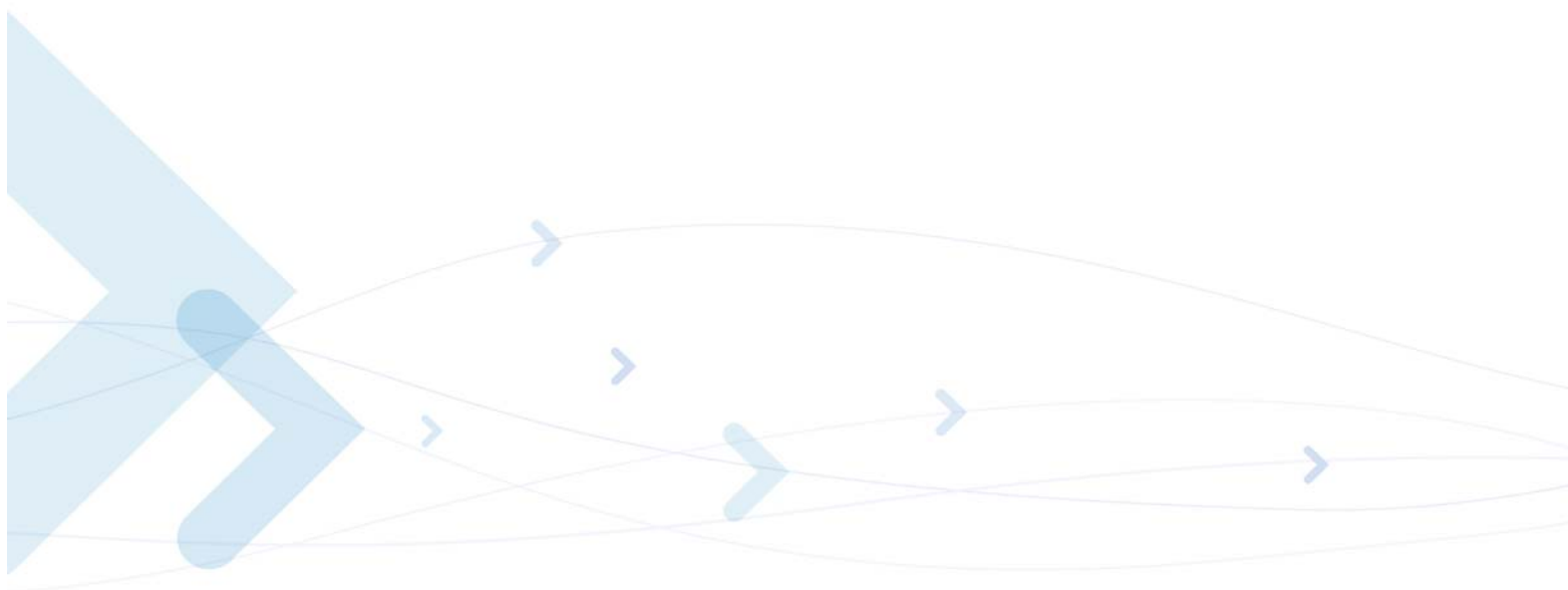


Technical Information



Motorola W24 Developer's Guide **AT+i Commands Reference Manual**

MAY 31, 2008
6802985C10-A

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

Notice

While reasonable efforts have been made to assure the accuracy of this document, Motorola, Inc. assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Motorola, Inc. reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Motorola, Inc. does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Motorola products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Motorola intends to announce such Motorola products, programming, or services in your country.

Copyrights

This instruction manual, and the Motorola products described in this instruction manual may be, include or describe copyrighted Motorola material, such as computer programs stored in semiconductor memories or other media. Laws in the United States and other countries preserve for Motorola and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Motorola and its licensors contained herein or in the Motorola products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Motorola. Furthermore, the purchase of Motorola products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Motorola, as arises by operation of law in the sale of a product.

Computer Software Copyrights

The Motorola and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Motorola and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the United States and other countries preserve for Motorola and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Motorola or other 3rd Party supplied SW computer programs contained in the Motorola products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Motorola or the 3rd Party SW supplier. Furthermore, the purchase of Motorola products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Motorola or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

Usage and Disclosure Restrictions

License Agreements

The software described in this document is the property of Motorola, Inc. and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Motorola, Inc.

High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Motorola and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

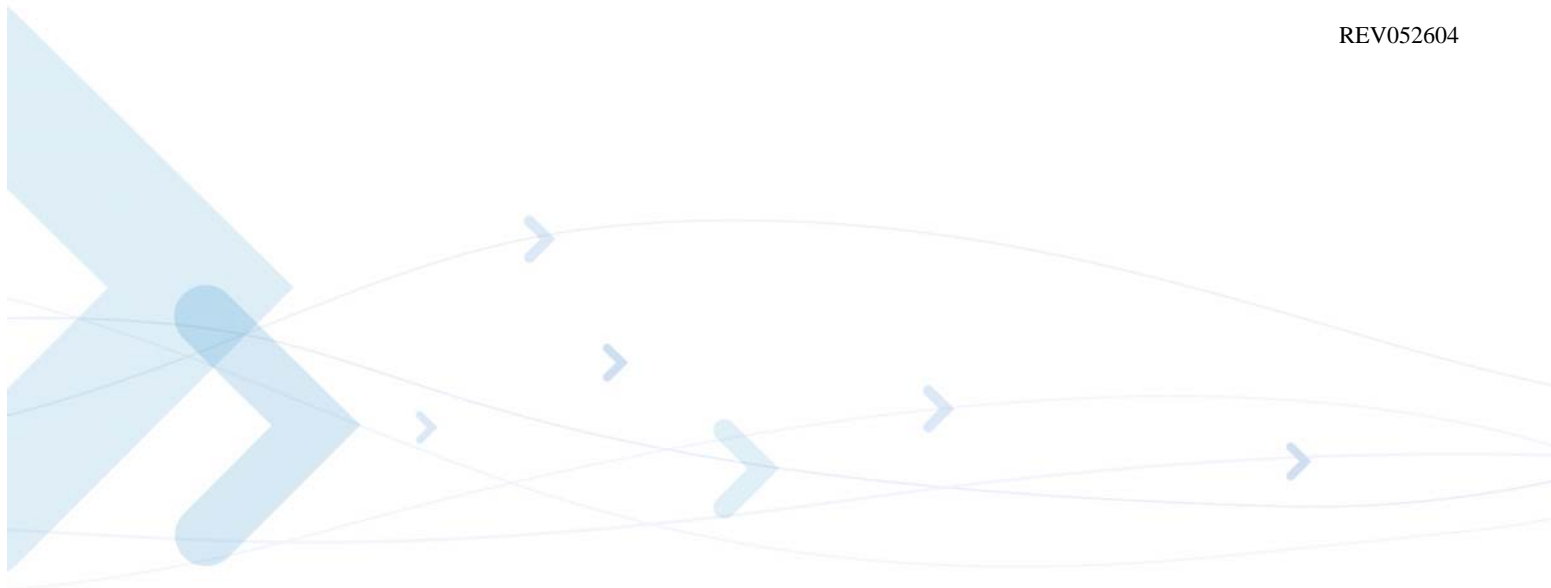
Trademarks



MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

©Copyright 2008 Motorola, Inc.

REV052604



Manual Scope	xiii
Target Audience	xiii
Manual Organization	xiii
Applicable Documents	xiii
Contact Us	xiv
Text Conventions	xiv
Manual Banner Definitions	xiv
Field Service	xv
General Safety	xv
Caring for the Environment	xvi
Limitation of Liability	xvii
Warranty Notification	xvii
How to Get Warranty Service?	xviii
Claiming	xviii
Conditions	xix
What is Not Covered by the Warranty	xix
Installed Data	xx
Out of Warranty Repairs	xx
Revision History	xxi
Chapter 1: Introduction to AT+i Commands	1-1
AT+i Commands Overview	1-1
AT+i Commands Format	1-1
Escape Code Sequence	1-2
Socket Command Abort	1-2
Flexible Host and Modem Interfaces	1-3
Auto Baud Rate Detection	1-3
High Speed USART	1-4
Entering Rescue Mode during Runtime	1-4
Internet Session Hang-up Procedure (Modem Only)	1-5
Modem Startup	1-5
Analog-to-Digital Converter	1-5
W24 Readiness Indication	1-6
Programming W24's Serial Number Into the Flash Memory	1-6
General AT+i Command Format	1-7
AT+i Commands by Category	1-7
AT+i Result Code Summary	1-12
Chapter 2: AT+i Commands Reference	2-1
Report Status	2-1
+i[!]R <i>Pi</i> - Report Status	2-1
Status Message Format	2-2
Connection	2-6
+iBDRA - Force W24 into Auto Baud Rate Mode	2-6
+iUP - Initiate Internet Session	2-6

+iTUP - Triggered Internet Session Initiation	2-7
+iDOWN - Terminate Internet Session	2-8
+iPING - Send a PING Request to a Remote Server	2-9
E-mail Send Commands	2-10
+iEMA - Accept ASCII-Coded Lines for E-Mail Send	2-10
+iEMB - Accept Binary Data for Immediate E-Mail Send	2-11
+iE* - Terminate Binary E-Mail	2-12
E-Mail Retrieve	2-13
+iRML - Retrieve Mail List	2-13
+iRMH - Retrieve Mail Header	2-13
+iRMM - Retrieve Mail Message	2-14
HTTP Client Interface	2-17
+iRLNK - Retrieve Link	2-17
+iSLNK - Submit a POST Request to a Web Server	2-18
SerialNET Mode Initiation	2-19
+iSNMD - Activate SerialNET Mode	2-19
Web Server Interface	2-21
+iWWW - Activate Embedded Web Server	2-21
+iWNXT - Retrieve Next Changed Web Parameter	2-21
File Transfer Protocol (FTP)	2-22
+i[@]FOPN - FTP Open Session	2-22
+iFDL - FTP Directory Listing	2-23
+iFDNL - FTP Directory Names Listing	2-23
+iFMKD - FTP Make Directory	2-24
+iFCWD - FTP Change Working Directory	2-24
+iFSZ - FTP File Size	2-25
+iFRCV - FTP Receive File	2-25
+iFSTO - FTP Open File for Storage	2-26
+iFAPN - FTP Open File for Appending	2-27
+iFSND - FTP Send File Data	2-27
+iFCLF - FTP Close File	2-28
+iFDEL - FTP Delete File	2-28
+iFCLS - FTP Close Session	2-29
Telnet Client.	2-30
+iTOPN - Telnet Open Session	2-30
+iTRCV - Telnet Receive Data	2-30
+iTSND - Telnet Send Data Line	2-30
+iTBSN[%] - Telnet Send a Byte Stream	2-31
+iTFSH[%] - Flush Telnet Socket's Outbound Data	2-32
+iTCLS - Telnet Close Session	2-32
Direct Socket Interface.	2-33
+iSTCP - Open and Connect a TCP Socket	2-33
+iSUDP - Open a Connectionless UDP Socket	2-33
+iLTCP - Open a TCP Listening Socket	2-34
+iLSST - Get a Listening Socket's Active Connection Status	2-35
+iSST - Get a Single Socket Status Report	2-35
+iSCS - Get a Socket Connection Status Report	2-36
+iSSND[%] - Send a Byte Stream to a Socket	2-37
+iSRCV - Receive a Byte Stream from a Socket's Input Buffer	2-38
+iGPNM - Get Peer Name for a Specified Socket	2-38
+iSDMP - Dump Socket Buffer	2-39
+iSFSH[%] - Flush Socket's Outbound Data	2-39
+iSCLS - Close Socket	2-40
Special Modem Commands	2-41

+iMCM - Issue Intermediate Command to Modem	2-41
Wireless LAN Mode	2-42
+iWLTR - Wireless LAN Transmission Rate	2-42
+iWLPW - Set WLAN Tx Power	2-43
+iWRFU - WLAN Radio Up	2-43
+iWRST - Reset WLAN Chipset	2-44
+iWLBW - WLAN B Mode	2-44
+iWLGW - WLAN G Mode	2-44
Roaming Mode	2-44
W24 Behavior Following a Hardware or Software Reset	2-45
W24 Behavior when AP Signal Becomes Weak	2-45
W24 Behavior in the Event of a Lost Link	2-45
Multiple SSIDs	2-46
W24 Power Save Mode	2-46
IP Registration	2-47
E-Mail Registration	2-47
Socket Registration	2-47
Web Server Registration	2-48
DHCP Client	2-49
DHCP Server	2-50
iRouter Mode	2-51
Introduction	2-51
Establishing iRouter Mode	2-51
Basic Routing	2-51
Configuring W24 when in iRouter Mode	2-52
Configuring W24 when in iRouter Mode	2-52
AT+i Interface to W24	2-52
Baud Rate Settings and Auto Baud Rate	2-53
iRouter and Power Save Mode	2-53
+iSTRR - Start Router	2-53
+iSTPR - Stop Router	2-54
Ad-Hoc Networks	2-55
Configuration	2-55
W24 Behavior in Ad-Hoc Mode	2-55
Automatic Scanning for Existing Ad-Hoc Networks	2-55
Creating a New Ad-Hoc Network	2-55
Joining an Existing Ad-Hoc Network	2-55
Merging Ad-Hoc Networks	2-56
Secure Socket Protocol	2-57
Establishing An SSL3/TLS1 Socket Connection	2-57
Sending and Receiving Data over An SSL3/TLS1 Socket	2-57
SSL3/TLS1 Handshake and Session Example	2-57
Secure FTP Session on W24	2-58
+iSSL - Secure Socket Connection Handshake	2-59
+i[@]FOPS - Secure FTP Open Session	2-59
Network Time Client	2-61
MIME Encapsulated E-Mail Messages	2-62
W24-Generated Binary Message Formats	2-62
MIME-Related AT+i Commands and Parameters	2-62
Binary Attachment Parameters	2-63
Defining a Textual Body for Binary Messages	2-63
MIME-Encapsulated E-Mail Message Format	2-64
Flow Control	2-66
Host -> W24 Software Flow Control	2-66

Software Flow Control Diagram in Binary E-Mail Send	2-67
Software Flow Control During a Socket Send	2-68
Software Flow Control Diagram in Socket Send	2-69
Host -> W24 Hardware Flow Control	2-70
Remote Firmware Update	2-71
Introduction	2-71
Updating Firmware from a Remote Server	2-71
+iRFU - Remote Firmware Update	2-72
W24 Parameter Update	2-73
Introduction	2-73
Remote Parameter File (RPF) Structure	2-73
Header Parameter Names and Values	2-73
Uploading a Parameters Update File to W24	2-74
W24 Embedded Web Server	2-75
Introduction	2-75
Features	2-75
Web Server Modes	2-75
The Application Website	2-76
Parameter Tags	2-76
W24 Configuration Mode	2-76
Host Interaction Mode	2-77
Website Creation, Packing, and Uploading	2-78
Manipulating Variables in the Application Website	2-79
Security and Restrictions	2-80
Parameter Update Error Handling	2-81
File Types Supported by W24's Web Server	2-81
W24 RAS Server	2-82
Introduction	2-82
RAS Parameters	2-82
RAS Theory of Operation	2-82
Auto PPP RAS Mode	2-83
SerialNET Mode.	2-83
Lost Carrier.	2-84
Restrictions	2-84
SerialNET Theory of Operation.	2-85
Introduction	2-85
SerialNET Mode	2-85
Server Devices	2-86
Client Devices	2-86
Automatic SerialNET Server Wake-Up Procedure	2-87
Transmit Packets	2-87
Completing a SerialNET Session	2-88
SerialNET Failed Connection	2-88
Local Serial Port Configuration	2-88
Activation Command	2-88
SerialNET over TELNET	2-89
Mode of Operation	2-89
RFC2217 Implementation	2-90
File Transfer Protocol (FTP) Theory of Operation	2-92
Introduction	2-92
W24 Family FTP Client Command Set	2-92
W24 FTP Client Operation Mode	2-92
FTP Command Socket	2-92
FTP Receive Flow	2-93

Telnet Client Operation	2-94
Secure Socket Protocol Theory of Operation	2-95
Introduction	2-95
Generating Certificates for Use with Servers	2-95
Using the OpenSSL Package to Create Certificates	2-95
Creating a Certificate Authority	2-96
Creating the CA Environment	2-96
Creating the Test CA Configuration File	2-96
Creating a Self-Signed Root Certificate	2-98
Signing a Certificate with a CA Certificate	2-98
Creating a Certificate Request	2-98
Using the Test CA to Issue the Certificate	2-99
Remote AT+i Service	2-100
Introduction	2-100
Remote AT+i Commands	2-100
Closing a Remote AT+i Session	2-100
Caveats and Restrictions	2-100
Nonvolatile Parameter Database	2-102
Parameter Descriptions	2-102
+iFD - Restore All Parameters to Factory Defaults	2-107
Operational Parameters	2-108
+iXRC - Extended Result Code	2-108
+iDMD - Modem Dial Mode	2-108
+iMIS - Modem Initialization String	2-109
+iMTYP - Set Type of Modem Connected to W24	2-110
+iWTC - Wait Time Constant	2-111
+iTTO - TCP Timeout	2-111
+iPGT - PING Timeout	2-112
+iMPS - Max PPP Packet Size	2-113
+iTTR - TCP Retransmit Timeout	2-113
+iBDRF - Define a Fixed Baud Rate on Host Connection	2-114
+iBDRM - Define a Fixed Baud Rate on W24<->Modem Connection	2-115
+iBDRD - Baud Rate Divider	2-115
+iAWS - Activate WEB Server Automatically	2-116
+iLATI - TCP/IP Listening Socket to Service Remote AT+i Commands	2-116
+iFLW - Set Flow Control Mode	2-118
+iCPF - Active Communications Platform	2-118
+iPSE - Set Power Save Mode	2-119
+iMRST - Turn the W24 Off	2-120
+iS102 - Define Delay after Wakeup before Sending Data	2-120
+iS100 - Define Wait Interval Between Wakeup Events	2-121
+iSDM - Service Disabling Mode	2-121
+iDF - IP Protocol 'Don't Fragment' Bit Value	2-122
+iCKSM - Checksum Mode	2-123
+iHIF - Host Interface	2-123
+iMIF - Modem Interface	2-124
+iADCL - ADC Level	2-125
+iADCD - ADC Delta	2-125
+iADCT - ADC Polling Time	2-126
+iADCP - ADC GPIO Pin	2-126
+iRRA - W24 Readiness Report Activation	2-127
+iRRHW - W24 Readiness Hardware Pin	2-128

ISP Connection Parameters	2-129
+iISPn - Set ISP Phone Number	2-129
+iATH - Set PPP Authentication Method	2-129
+iUSRN - Define Connection User Name	2-130
+iPWD - Define Connection Password	2-130
+iRDL - Number of Times to Redial ISP	2-131
+iRTO - Delay Period between Redials to ISP	2-132
Server Profile Parameters	2-132
+iLVS - 'Leave on Server' Flag	2-132
+iDNSn - Define Domain Name Server IP Address	2-134
+iSMTP - Define SMTP Server Name	2-135
+iSMA - SMTP Authentication Method	2-136
+iSMU - Define SMTP Login User Name	2-136
+iSMP - Define SMTP Login Password	2-137
+iPOP3 - Define POP3 Server Name	2-137
+iMBX - Define POP3 Mailbox Name	2-138
+iMPWD - Define POP3 Mailbox Password	2-139
+iNTSn - Define Network Time Server	2-139
+NTOD - Define Network Time-of-Day Activation Flag	2-140
+iGMTO - Define Greenwich Mean Time Offset	2-141
+iDSTD - Define Daylight Savings Transition Rule	2-141
+iPDSn - Define PING Destination Server	2-142
+iPFR - PING Destination Server Polling Frequency	2-142
+iUFn - User Fields and Macro Substitution	2-143
Email Format Parameters	2-144
+iXFH - Transfer Headers Flag	2-144
+iHDL - Limit Number of Header Lines	2-144
+iFLS - Define Filter String	2-145
+iDELF - Email Delete Filter String	2-146
+iSBJ - Email Subject Field	2-146
+iTOA - Define Primary Addressee	2-147
+iTO - Email 'To' Description/Name	2-147
+iREA - Return Email Address	2-148
+iFRM - Email 'From' Description/Name	2-149
+iCCn - Define Alternate Addressee <n>	2-149
+iMT - Media Type Value	2-150
+iMST - Media Subtype String	2-150
+iFN - Attachment File Name	2-151
HTTP Parameters	2-152
+iURL - Default URL Address	2-152
+iCTT - Define Content Type Field in POST Request	2-152
+iWPWD - Password for Application Website Authentication	2-153
RAS Server Parameters	2-154
+iRAR - RAS RINGs	2-154
+iRAU - Define RAS Login User Name	2-154
+iRAP - Password for RAS Authentication	2-155
LAN Parameters	2-156
+iMACA - MAC Address of W24	2-156
+iDIP - W24 Default IP Address	2-156
+iIPA - Active IP Address	2-157
+iIPG - IP Address of the Gateway	2-157
+iSNET - Subnet Address	2-158

Wireless LAN Parameters	2-159
+iWLCH - Wireless LAN Communication Channel	2-159
+iWLSI - Wireless LAN Service Set Identifier	2-159
+iWLWM - Wireless LAN WEP Mode	2-160
+iWLKI - Wireless LAN Transmission WEP Key Index	2-160
+iWLKn - Wireless LAN WEP Key Array	2-161
+iWLPS - Wireless LAN Power Save	2-162
+iWLPP - Personal Shared Key Pass-Phrase	2-162
+iWROM - Enable Roaming in WiFi	2-163
+iWPSI - Periodic WiFi Scan Interval	2-163
+iWSRL - SNR Low Threshold	2-164
+iWSRH - SNR High Threshold	2-164
+iWSIn - Wireless LAN Service Set Identifier Array	2-165
+iWPPn - Pre-Shared Key Passphrase Array	2-166
+iWKYn - Wireless LAN WEP Key Array	2-166
+iWSTn - Wireless LAN Security Type Array	2-167
+iWSEC - Wireless LAN WPA Security	2-168
IP Registration Parameters	2-168
+iRRMA - IP Registration Mail Address	2-168
+iRRSV - IP Registration Host Server Name	2-169
+iRRWS - IP Registration Web Server	2-170
+iRRRL - IP Registration Return Link	2-170
+iHSTN - W24 Network Host Name	2-171
SerialNET Mode Parameters	2-172
+iHSRV +iHSRn - Host Server Name/IP	2-172
+iHSS - Assign Special Characters to Hosts	2-173
+iDSTR - Define Disconnection String for SerialNET Mode	2-173
+iLPRT - SerialNET Device Listening Port	2-174
+iMBTB - Max Bytes To Buffer	2-174
+iMTTF - Max Timeout to Socket Flush	2-175
+iFCHR - Flush Character	2-175
+iMCBF - Maximum Characters before Socket Flush	2-176
+iIATO - Inactivity Timeout	2-177
+iSNSI - SerialNET Device Serial Interface	2-178
+iSTYP - SerialNET Device Socket Type	2-179
+iSNRD - SerialNET Device Re-Initialization Delay	2-180
+iSPN - SerialNET Server Phone Number	2-181
+iSDT - SerialNET Dialup Timeout	2-182
+iSWT - SerialNET Wake-Up Timeout	2-183
+iPTD - SerialNET Packets to Discard	2-184
Remote Firmware Update Parameters	2-184
+iUEN - Remote Firmware Update Flag	2-184
+iUSRV - Remote Firmware Update Server Name	2-185
+iUUSR - Remote Firmware Update FTP User Name	2-186
+iUPWD - Remote Firmware Update FTP User Password	2-186
Remote Parameter Update	2-187
Secure Socket Protocol Parameters	2-188
+iCS - Define the SSL3/TLS Cipher Suite	2-188
+iCA - Define SSL3/TLS Certificate Authority	2-189
+iCERT - Define SSL3/TLS1 Certificate	2-190
+iPKEY - Define W24's Private Key	2-190
DHCP Server Parameters	2-192
+iDPSZ - DHCP Server Pool Size	2-192
+iDSLT - DHCP Server Lease Time	2-192

iRouter Parameters 2-193

+iARS - Automatic Router Start. 2-193

Appendix A: MIME Content Types and SubtypesA-1

Appendix B: Sample Parameter Update FileB-1

Appendix C: NIST Time ServersC-1

Appendix D: Use CasesD-1

 Use Case - Host ModeD-1

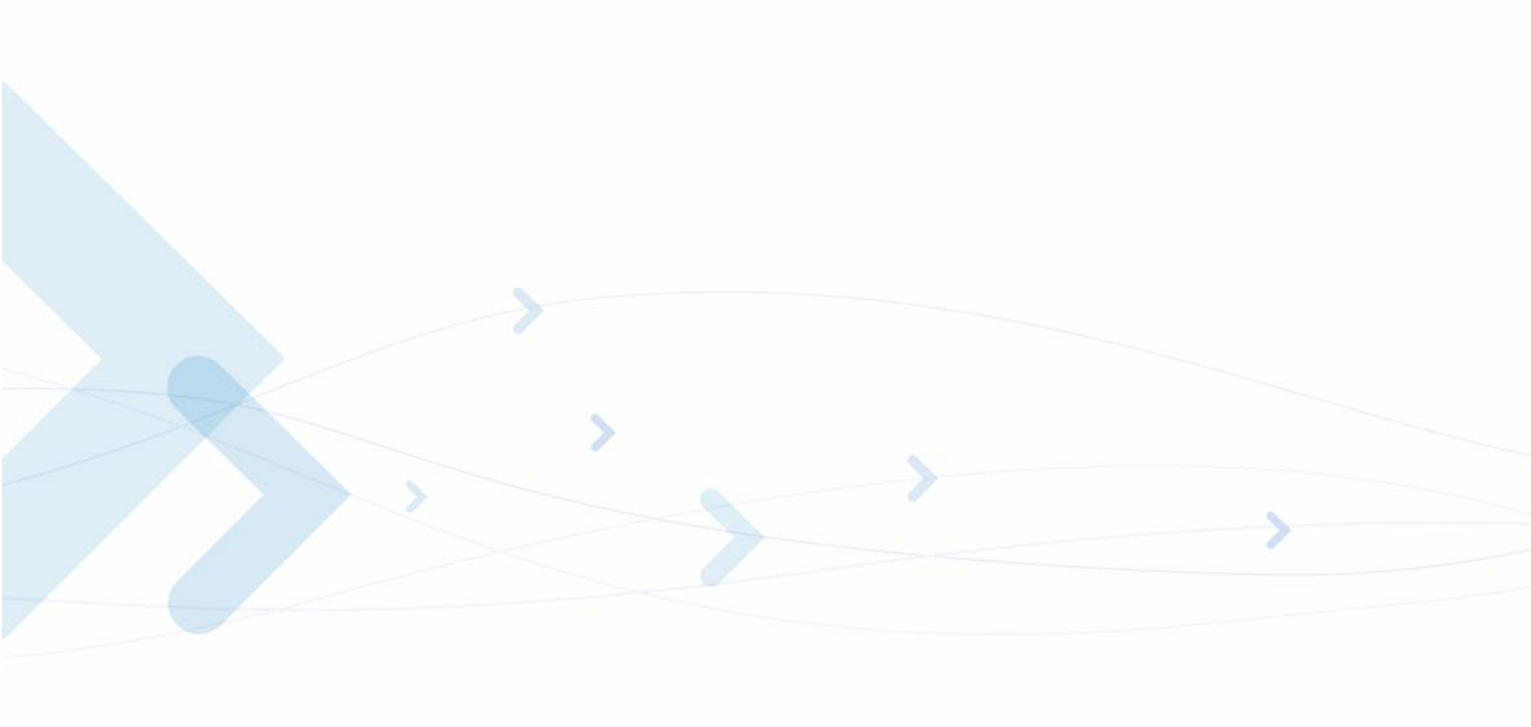
 The mode configurationD-1

 Use Case - Gateway/Router Mode.D-4

 Routing inside of W24D-5

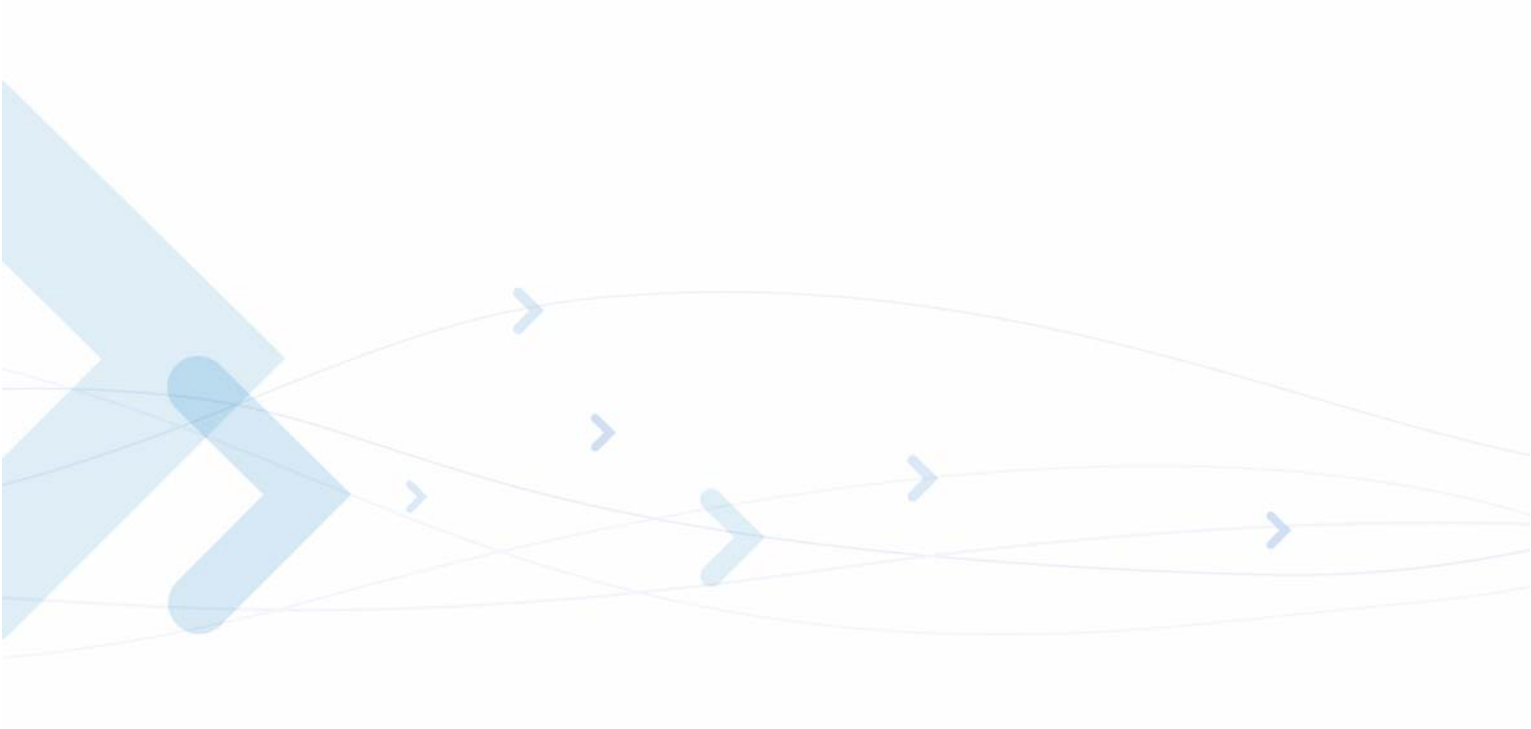
 Use Case - Integrated Host (Java) ModeD-6

Index



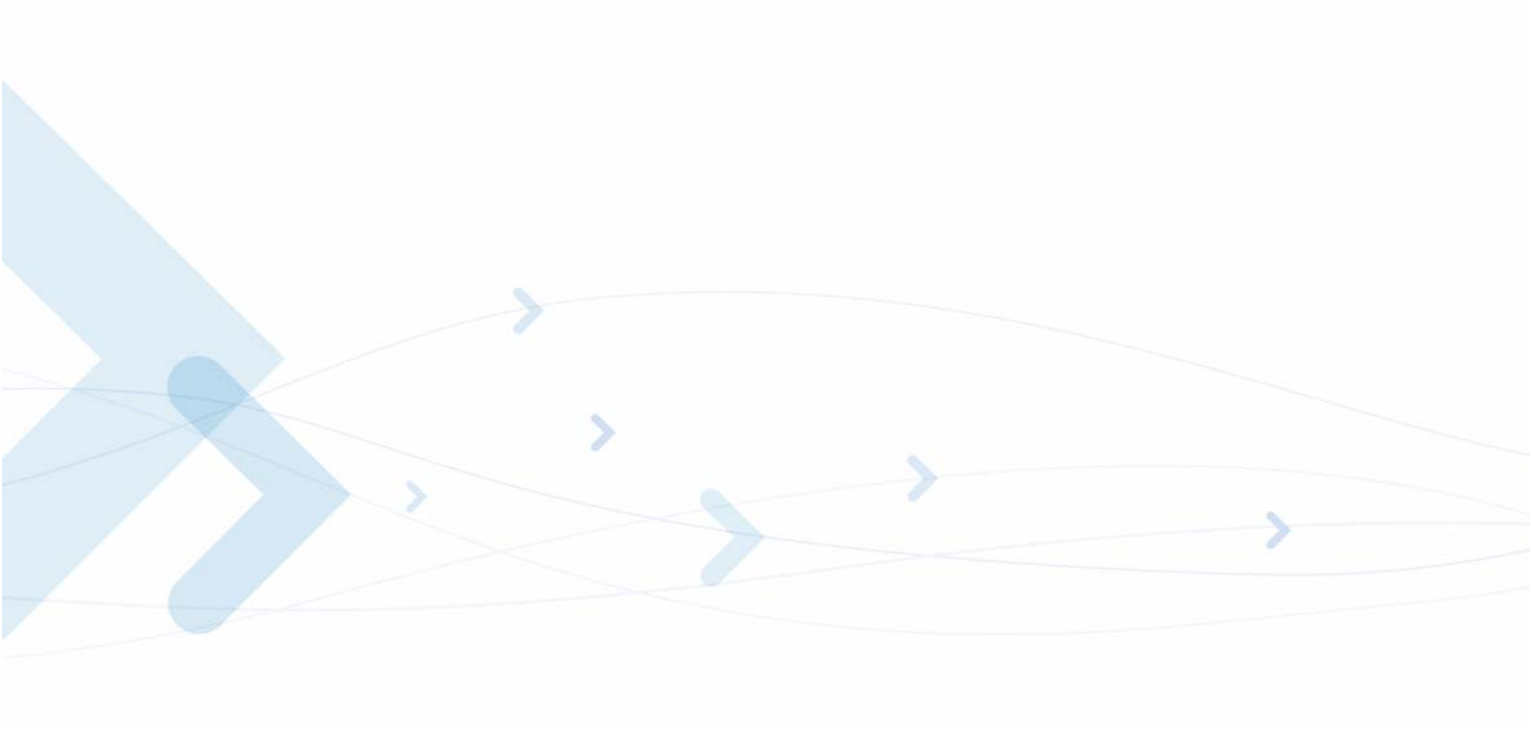
List of Figures

2-1	E-Mail Receive (RMM) Flow Diagram	2-16
2-2	Configuring W24 when in iRouter Mode	2-52
2-3	AT+i Interface to W24	2-53
2-4	Software Flow Control in Binary E-Mail Send	2-67
2-5	Software Flow Control in Socket Send	2-69
2-6	Minimum Hardware Flow Control Connections	2-70
2-7	W24 Web Server Modes	2-76
2-8	FTP Receive Flowchart	2-93
2-9	Lines Behavior During Wakeup Event	2-120
D-1	External Terminal/Host Data Mode	D-1
D-2	Gateway/Router Data Mode	D-5
D-3	Integrated Host (Java) Mode	D-6



List of Tables

1-1	AT+i Commands by Category	1-7
1-2	AT+i Result Code Summary	1-12
2-1	Report Status Message Format	2-2
2-2	Server Names Acquired from DHCP Server	2-49
2-3	Binary Attachment Parameters	2-63
2-4	Software Flow Control Characters	2-66
2-5	Header Parameter Names and Values	2-73
2-6	Nonvolatile Parameter Database	2-102
C-1	List of NIST Time Servers	C-1



Preface

Manual Scope

This manual introduces the W24 AT+i commands, and describes how software developers can use these commands to communicate with the W24 device, and to create software applications that communicate with the W24 using these commands.

Note: The integrator should read the corresponding SW release notes for the W24 version he is using to get information about differences from this manual.

Target Audience

This manual is intended for software developers who communicate with the W24 device using the AT+i commands, and create applications to communicate with the W24 device using the AT+i commands.

Manual Organization

This manual contains the following chapters:

- “[Preface](#)” provides a scope for this manual, document convention, safety instructions and a liability notification.
- “[Chapter 1: Introduction to AT+i Commands](#)” gives an overview of the AT+i commands.
- “[Chapter 2: AT+i Commands Reference](#)” provides a reference to all available AT+i commands, including examples, where relevant.
- “[Appendix A: MIME Content Types and Subtypes](#)” provides MIME content types and subtypes.
- “[Appendix B: Sample Parameter Update File](#)” provides sample parameter update file.
- “[Appendix C: NIST Time Servers](#)” provides a list of NIST time servers.
- “[Appendix D: Use Cases](#)” provides some use cases.

Applicable Documents

- W24 Module Hardware Description – 6802984C95
- W24 Developer’s Kit – 6802985C05

Contact Us

We at Motorola want to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

For general contact, technical support, report documentation errors and to order manuals, use this email address:

M2M.CustomerCare@motorola.com

Motorola appreciates feedback from the users of our information.

Text Conventions

The following special paragraphs are used in this guide to point out information that must be read. This information may be set-off from the surrounding text, but is always preceded by a bold title in capital letters:

Note

Note: Presents additional, helpful, noncritical information that you can use.

Warning

Warning: Presents information to warn you of a potentially hazardous situation in which there is a possibility of personal injury.

Important

Important: Presents information to help you avoid an undesirable situation or provides additional information to help you understand a topic or concept.

Caution

Caution: Presents information to identify a situation in which damage to software, stored data, or equipment could occur, thus avoiding the damage.

Manual Banner Definitions

A banner text in the page footer under the book title (for example, **Preliminary** or **FOA**) indicates that some information contained in the manual is not yet approved for general customer use.

Field Service

For Field Service requests, use this email address:

M2M.Customer@motorola.com

General Safety

Remember!. . . safety depends on you!

The following general safety precautions must be observed during all phases of operation, service, and repair of the equipment described in this manual. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola, Inc. assumes no liability for the customer's failure to comply with these requirements. The safety precautions listed below represent warnings of certain dangers of which we are aware. You, as the user of this product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.

Ground the instrument

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. If the equipment is supplied with a three-conductor AC power cable, the power cable must be either plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter. The three-contact to two-contact adapter must have the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable must meet International Electrotechnical Commission (IEC) safety standards.

Note: Refer to *“Grounding Guideline for Cellular Radio Installations”*—Motorola part no. 68P081150E62.

Do not operate in an explosive atmosphere

Do not operate the equipment in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

Do not service or adjust alone

Do not attempt internal service or adjustment unless another person, capable of rendering first aid is present.

Keep away from live circuits

Operating personnel must:

- not remove equipment covers. Only Factory Authorized Service Personnel or other qualified maintenance personnel may remove equipment covers for internal subassembly, or component replacement, or any internal adjustment
- not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed
- always disconnect power and discharge circuits before touching them

Do not substitute parts or modify equipment

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of equipment. Contact Motorola Warranty and Repair for service and repair to ensure that safety features are maintained.

Dangerous procedure warnings

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions that you deem necessary for the operation of the equipment in your operating environment.

Warning example:

Warning: Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

Caring for the Environment

The following information is provided to enable regulatory compliance with the European Union (EU) Directive [2002/96/EC Waste Electrical and Electronic Equipment \(WEEE\)](#) when using Motorola equipment in EU countries.

Disposal of Motorola equipment in EU countries



Please do not dispose of Motorola equipment in landfill sites.

In the EU, Motorola in conjunction with a recycling partner will ensure that equipment is collected and recycled according to the requirements of EU environmental law.

Please contact the Customer Network Resolution Center (CNRC) for assistance. The 24 hour telephone numbers are listed at

<http://mynetworksupport.motorola.com>

Select **Customer Network Resolution Center** contact information.

Alternatively if you do not have access to CNRC or the internet, contact the Local Motorola Office.

Disposal of Motorola equipment in non-EU countries

In non-EU countries, dispose of Motorola Networks equipment in accordance with national and regional regulations.

RoHS Compliance

The W24 product meets the European Union directive for RoHS compliance.

The RoHS compliance is subject to a declaration of conformity that may be viewed upon request.

Limitation of Liability

The Products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body; in other applications intended to support or sustain life; for the planning, construction, maintenance, operation or use of any nuclear facility; for the flight, navigation, communication of aircraft or ground support equipment; or in any other application in which the failure of the Product could create a situation where personal injury or death may occur. If CUSTOMER should use any Product or provide any Product to a third party for any such use, CUSTOMER hereby agrees that MOTOROLA is not liable, in whole or in part, for any claims or damages arising from such use, and further agrees to indemnify and hold MOTOROLA harmless from any claim, loss, cost or damage arising from such use.

EXCEPT AS SPECIFICALLY STATED ABOVE, THE PRODUCTS ARE PROVIDED "AS IS" AND MOTOROLA MAKES NO OTHER WARRANTIES EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE REGARDING THE PRODUCTS. MOTOROLA SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING OR USAGE OF TRADE.

Under no circumstances shall MOTOROLA be liable to CUSTOMER or any other party for any costs, lost revenue or profits or for any other special, incidental or consequential damages, even if MOTOROLA has been informed of such potential loss or damage. And in no event shall MOTOROLA's liability to CUSTOMER for damages of any nature exceed the total purchase price CUSTOMER paid for the Product at issue in the dispute, except direct damages resulting from patent and/or copyright infringement, which shall be governed by the "INDEMNITY" Section of this Agreement.

The preceding states MOTOROLA's entire liability for MOTOROLA's breach or failure to perform under any provision of this Agreement.

Warranty Notification

Motorola guarantees to you, the original purchaser, the OEM module and accessories which you have purchased from an authorized Motorola dealer (the "Products"), to be in conformance with

the applicable Motorola specifications current at the time of manufacture for a term of [1] year from date of purchase of the Product(s) (Warranty Term).

You must inform Motorola of the lack of conformity to the applicable specifications of any of the Products within a period of two (2) months from the date on which you detect a defect in material, workmanship or lack of conformity and in any event within a term not to exceed the Warranty Term, and must immediately submit the Product for service to Motorola's Authorized Repair or Service Center. Motorola shall not be bound by Product related statements not directly made by Motorola nor any warranty obligations applicable to the seller.

A list of the Motorola Call Center numbers is enclosed with this Product.

During the Warranty term, Motorola will, at its discretion and without extra charge, as your exclusive remedy, repair or replace your Product which does not comply with this warranty; or failing this, to reimburse the price of the Product but reduced to take into account the use you have had of the Product since it was delivered. This warranty will expire at the end of the Warranty Term.

This is the complete and exclusive warranty for a Motorola OEM module and accessories and in lieu of all other warranties, terms and conditions, whether express or implied.

Where you purchase the product other than as a consumer, Motorola disclaims all other warranties, terms and conditions express or implied, such as fitness for purpose and satisfactory quality.

In no event shall Motorola be liable for damages nor loss of data in excess of the purchase price nor for any incidental special or consequential damages* arising out of the use or inability to use the Product, to the full extent such may be disclaimed by law.

This Warranty does not affect any statutory rights that you may have if you are a consumer, such as a warranty of satisfactory quality and fit for the purpose for which products of the same type are normally used under normal use and service, nor any rights against the seller of the Products arising from your purchase and sales contract.

(*)including without limitation loss of use, loss of time, loss of data, inconvenience, commercial loss, lost profits or savings.

How to Get Warranty Service?

In most cases the authorized Motorola dealer which sold and/or installed your Motorola OEM module and original accessories will honor a warranty claim and/or provide warranty service. Alternatively, for further information on how to get warranty service please contact the Motorola M2M Data Module Customer Support Center.

Claiming

In order to claim the warranty service you must return the OEM module and/or accessories in question to Motorola's Authorized Repair or Service Center in the original configuration and packaging as supplied by Motorola. Please avoid leaving any supplementary items like SIM cards. The Product should also be accompanied by a label with your name, address, and telephone number; name of operator and a description of the problem.

In order to be eligible to receive warranty service, you must present your receipt of purchase or a comparable substitute proof of purchase bearing the date of purchase. The module should also clearly display the original Motorola Serial Number (MSN). Such information is contained with the Product.

You must ensure that all and any repairs or servicing is handled at all times by a Motorola Authorized Service Center in accordance with the Motorola Service requirements.

In some cases, you may be requested to provide additional information concerning the maintenance of the Products by Motorola Authorized Service Centers only, therefore it is important to keep a record of any previous repairs, and make them available if questions arise concerning maintenance.

Conditions

This warranty will not apply if the type or serial numbers on the Product has been altered, deleted, duplicated, removed, or made illegible. Motorola reserves the right to refuse free-of-charge warranty service if the requested documentation can not be presented or if the information is incomplete, illegible or incompatible with the factory records.

Repair, at Motorola's option, may include reflashing of software, the replacement of parts or boards with functionally equivalent, reconditioned or new parts or boards. Replaced parts, accessories, batteries, or boards are warranted for the balance of the original warranty time period. The Warranty Term will not be extended. All original accessories, batteries, parts, and OEM module equipment that have been replaced shall become the property of Motorola. Motorola does not warrant the installation, maintenance or service of the products, accessories, batteries or parts.

Motorola will not be responsible in any way for problems or damage caused by any ancillary equipment not furnished by Motorola which is attached to or used in connection with the Products, or for operation of Motorola equipment with any ancillary equipment and all such equipment is expressly excluded from this warranty.

When the Product is used in conjunction with ancillary or peripheral equipment not supplied by Motorola, Motorola does not warrant the operation of the Product/peripheral combination and Motorola will not honor any warranty claim where the Product is used in such a combination and it is determined by Motorola that there is no fault with the Product. Motorola specifically disclaims any responsibility for any damage, whether or not to Motorola equipment, caused in any way by the use of the OEM module, accessories, software applications and peripherals (specific examples include, but are not limited to: batteries, chargers, adapters, and power supplies) when such accessories, software applications and peripherals are not manufactured and supplied by Motorola.

What is Not Covered by the Warranty

This warranty is not valid if the defects are due to damage, misuse, tampering, neglect or lack of care and in case of alterations or repair carried out by unauthorized persons.

The following are examples of defects or damage not covered by this product warranty

1. Defects or damage resulting from use of the Product in other than its normal and customary manner.
2. Defects or damage from misuse, access to incompatible sources, accident or neglect.
3. Defects or damage from improper testing, operation, maintenance, installation, adjustment, unauthorized software applications or any alteration or modification of any kind.
4. Breakage or damage to antennas unless caused directly by defects in material or workmanship.

5. Products disassembled or repaired other than by Motorola in such a manner as to adversely affect performance or prevent adequate inspection and testing to verify any warranty claim.
6. Defects or damage due to range, coverage, availability, grade of service, or operation of the Wi-Fi Provider.
7. Defects or damage due to moist, liquid or spills of food.
8. Control unit coil cords in the Product that are stretched or have the modular tab broken.
9. All plastic surfaces and all other externally exposed parts that are scratched or damaged due to customer normal use.

Depending on operating conditions and your usage habits, wear and tear might take place of components including mechanical problems related to Product housing, paint, assembly, sub-assemblies, displays and keyboards and any accessories which are not part of the Product's in-box configuration. The rectification of faults generated through wear and tear and the use of consumable items like batteries beyond their Optimum Performance Time as indicated in the product manual is considered to be your responsibility and therefore Motorola will not provide the free Warranty repair service for these items.

Installed Data

Please make and retain a note of all data you have inserted into your product. For example names, addresses, phone numbers, user and access codes, notes etc. before submitting your product for a warranty service as such data may be deleted or erased as part of the repair or service process.

Please note if you have downloaded material onto your product, for example ring tones, ring tunes, screensavers, wallpaper, games, etc. These may be deleted or erased as part of the repair process or testing process. Motorola shall not be responsible for such matters. The repair or testing process should not affect any such material that was installed by Motorola on your product as a standard feature.

Out of Warranty Repairs

Out of warranty HW repairs are not applicable for W24. Therefore, the defective unit shall be scrapped.

Revision History

Manual Number

6802985C10-A

Manual Title

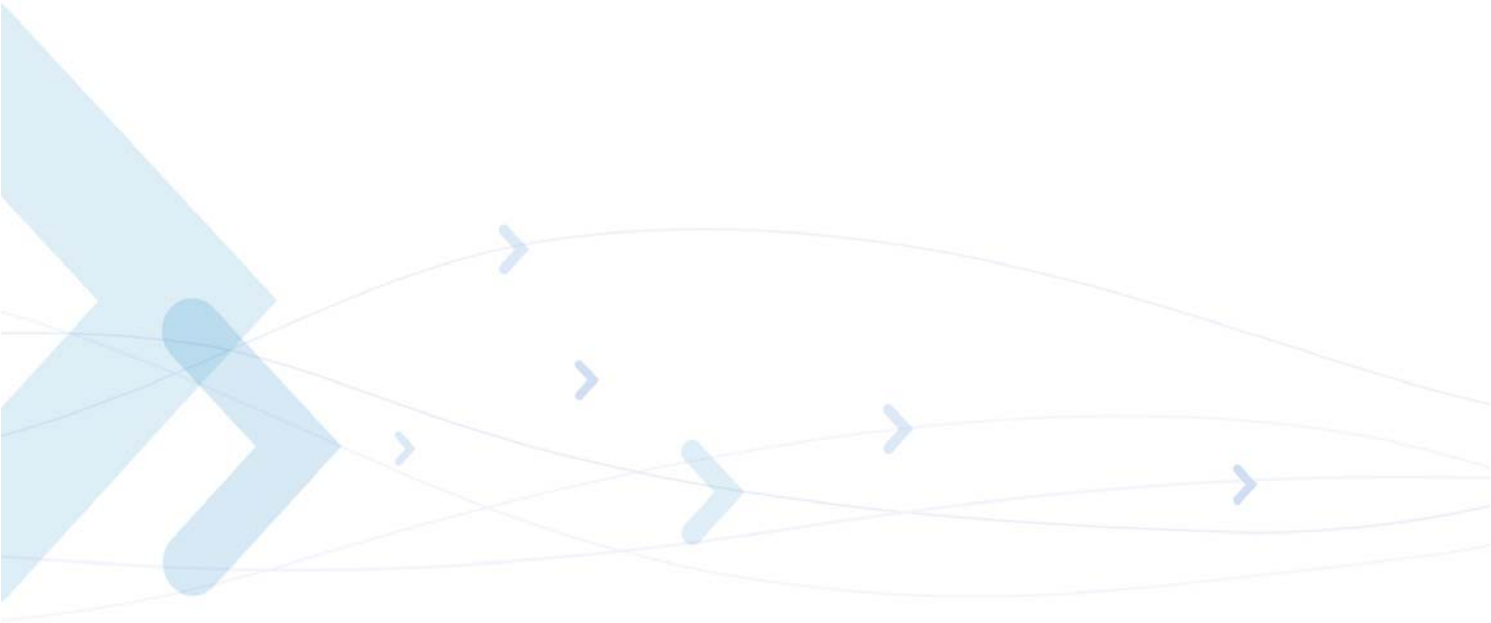
W24 Developer's Guide: AT+i Commands Reference Manual

Version Information

The following table lists the manual version, date of version, and remarks about the version.

Revision History

Version	Date Issue	Remarks
A	May 31, 2008	Initial Release



Chapter 1: Introduction to AT+i Commands

AT+i Commands Overview

AT+i commands are an extension to the basic AT commands set. They are parsed and acted upon by W24.

Note: The terms 'Command' and 'Parameter' are used throughout this document to refer to AT+i commands as follows:

- A 'Command' is an AT+i command that executes an operation
- A 'Parameter' is an AT+i command that sets/gets a parameter value in database

W24 in dial-up mode only: When W24 is in COMMAND mode, basic AT commands and raw data (not prefixed by AT+i) are transparently transferred to the underlying modem DCE (Digital Communications Equipment), where they are serviced. When transferring data transparently to the DCE, the hardware flow control signals (CTS, RTS, DTR and DSR) are mirrored across the W24, unless disabled by the FLW parameter. AT and AT+i commands may be issued intermittently. During an Internet session, when W24 is online, an AT command can be sent to the modem using the AT+iMCM command.

The ASCII ISO 646 character set (CCITT T.50 International Alphabet 5, American Standard Code for Information Interchange) is used for issuing commands and responses.

Only the low-order 7 bits of each character are used for commands or parameters; the high-order bit is ignored. Upper-case characters are equivalent to lower-case ones.

AT+i Commands Format

An AT+i command line is a string of characters sent from the host to the W24 while it is in command state. The command line has a prefix, a body, and a terminator. Each command must begin with the character sequence AT+i and terminated by a carriage return (<CR>). Commands can be entered either in upper-case or lower-case.

W24 in dial-up-mode only: Commands that do not begin with the AT+i prefix are transferred to the underlying DCE, where they are parsed and acted upon. DCE responses are transparently returned to the host.

The AT+i command body is restricted to printable ASCII characters (032-126). The command terminator is the ASCII <CR> character. The command line interpretation begins upon receipt of the carriage return character. An exception to this rule are the AT+iEMB, AT+iSSND, AT+iTBSN and AT+iFSND commands.

When ECHO is enabled, the <CR> character is echoed as a two-character sequence: <CR><LF> (Carriage Return+Line Feed).

Characters within the AT+i command line are parsed as commands with associated parameter values.

The W24 supports editing of command lines by recognizing a backspace character. When ECHO is enabled, the W24 responds to receipt of a backspace by echoing a backspace character, a space character, and another backspace. When ECHO is disabled, backspace characters are treated as data characters without any further processing.

If a syntax error is found anywhere in a command line, the remainder of the line is ignored and the I/ERROR result code returned.

An AT+i command is accepted by W24 once the previous command has been fully executed, which is normally indicated by the return of an appropriate result code.

Due to the fact that W24 is intended for Machine-to-Machine applications, only limited parsing is performed on AT+i commands it receives from the host. The following restrictions apply:

- When setting parameters to values larger than the 65535 limit, the values is accepted as modulo 65535.
- The validity of input IP addresses is not checked.
- Illegal numbers, for example, 0.5 or 1.5 are not checked for validity.

Escape Code Sequence

While the W24 is in Internet mode attending to Internet communications, it is possible to break into the communications and abort the Internet mode in an orderly manner. This is achieved by sending the W24 a sequence of three ASCII '+' characters ('+++') after a half second silence period. In response to this, the W24:

- Shuts down Internet communications.
- Terminates data transmission to the host.
- Performs a software reset.
- Responds with an I/ERROR(056) message.
- Returns to command mode.

A maximum delay of 10msec may elapse from the time the '+++' escape sequence is sent until W24 cuts-off transmission to the host. The interrupted Internet activity is not completed. Nevertheless, this is considered to comprise a session. Thus, parameters set with the '~' character are restored to their permanent value.

Socket Command Abort

While the W24 is in Internet mode, during a TCP or UDP socket operation, it is possible to override W24's normal timeout procedure and abort the current socket operation in an orderly manner. This is achieved by sending the W24 a sequence of three ASCII '-' characters ('---') following a half second silence period. The socket commands to which this applies are: STCP, SUDP, SSND, and SFSH. When W24 detects the socket abort command, it aborts the last socket command and returns an I/ERROR following the STCP and SUDP commands, or I/OK during an SSND or SFSH command.

Flexible Host and Modem Interfaces

The flexible host and modem interfaces feature enables users to select the interface through which W24 accepts AT+i commands from the host processor, as well as the interface through which AT+i commands are sent to a dialup or cellular modem.

Available host interfaces are:

- USART0
- USART1
- USART2
- USB Device (identifies itself as a CDC device)
- USB Host (supports only USB Modem class)

Available modem interfaces are:

- USART0
- USART1
- USART2
- USB Device
- USB Host

As a USB host/device, W24 supports the Full-Speed USB standard (12Mbps).

Host-to-W24 interface is selected by setting the value of the Host Interface (HIF) parameter. Any value from 1 to 5 specifies a certain choice of interface, while a 0 value specifies automatic interface detection. In automatic interface detection mode, the first character sent from the host over one of the supported interfaces sets the host interface to be used throughout that session until the next W24 power cycle.

When automatic host interface detection mode is enabled, a host is connected to one of the USARTs, and the Host Fixed Baud Rate (BDRF) parameter is set to 'a' (automatic baud rate detection), the first character the host has to send to W24 in order to trigger detection must be an 'a' or 'A'. If BDRF is set to a fixed baud rate, any character sent from the host triggers automatic host interface detection.

In a similar fashion, a W24-to-modem interface can be selected using the Modem Interface (MIF) parameter, except that automatic modem interface detection is not available.

Note that any changes to the HIF and MIF parameters take effect only after the following W24 power-up. Also note that W24 cannot be operated in SerialNET mode when the HIF parameter is set to automatic mode. Sending an SNMD command (activate SerialNET mode) with HIF set to automatic mode will result in an error message I/ERROR (122). In addition, any feature that requires setting a fixed baud rate requires setting a fixed host interface, as well.

Hardware flow control is supported on USART0 and USART1 only. Hardware signal mirroring is enabled only if the host and modem interfaces are set to either USART0 or USART1. See description of Bit 2 of the FLW parameter.

Auto Baud Rate Detection

W24 supports auto baud rate detection on the host serial communications line. After power-up, W24 enters auto baud mode when the BDRF parameter is set to the value 'a'. The AT+iBDRA command forces W24 into auto baud mode while it is already in operation.

In auto baud mode, W24 expects an A or a character. This is usually the first character sent, since in command mode a meaningful command is always prefixed by AT+i.

The host may send an a or A to the W24 to allow it to determine the host's baud rate. It may also send a complete AT+i command. In any case, W24 detects the A or a character, determines the correct baud rate, and configures its serial channel during the stop bit. Thus, the next character is received by the serial port at the correct baud rate. The A itself is retained as well. W24 supports auto baud rate detection for the following baud rates: 2400, 4800, 9600, 19200, 38400, 57600, and 115200.

When the BDRF parameter contains a fixed baud rate, W24 initializes to the specified baud rate without entering auto baud rate mode. Commands issued by the host must be sent using that baud rate in order to be recognized. In this case, W24 can be forced into auto baud rate mode, refer to [“Entering Rescue Mode during Runtime”](#).

W24 dial-up mode only: When the BDRM parameter is set to an a value, W24 assumes the attached modem has the auto baud rate feature. Once the host<>W24 baud rate is determined, the W24<>modem baud rate is set to the same rate. Any other BDRM value is used as a fixed baud rate to the modem.

High Speed USART

Very high baud rates, up to 3Mbps, can be reached between host and W24 via one of W24's USARTs. The BDRD parameter acts as baud rate divider. When set to '0', W24 sets its host USART baud rate according to the value of the BDRF parameter. When set to any value in the range 1-255, it divides the maximum supported baud rate - 3Mbps - by that value. The quotient of this division is set as the host baud rate, and the value of BDRF is ignored. For example, if BDRD is set to 2, then the host baud rate will be $3\text{Mbps} / 2 = 1.5\text{Mbps}$.

If the W24<>modem interface is a USART, BDRD is set to any value other than '0', and the modem baud rate is set to Auto (BDRM='a'), then the modem baud rate will be set to a fixed value of 115,200bps.

In SerialNET mode, you can specify that host<>W24 baud rate over USART be determined by the BDRD parameter. You do so by setting the first field of the SNSI parameter (<baud>) to '0'.

Entering Rescue Mode during Runtime

The MSEL (Mode Select) input signal of the W24 can be used for entering W24 into Rescue mode.

If MSEL is pulled low (logical 0) for more than 5 seconds during runtime, W24 waits until MSEL is pulled high (logical 1), performs a software reset and restarts in Rescue mode. In Rescue mode, W24 performs the following operations:

- If in SerialNET mode - W24 exits SerialNET mode (changes SNMD value to 0).
- If serial baud rate (BDRF or BDRD) is set to a fixed value - W24 forces auto baud rate detection. BDRF/BDRD value will be used again upon the next power-up.
- If Always Online mode is defined (TUP=2), or Automatic Router Start is enabled (ARS=1) - W24 bypasses this mode, which means that W24 does not attempt to go online until the next software or hardware reset.
- If the Host Interface parameter (HIF) is set to a fixed interface, it is forced into auto host interface detection mode (HIF=0).

Internet Session Hang-up Procedure (Modem Only)

Upon completion of a dial-up Internet session, the W24 automatically executes a modem hang-up procedure:

- The DTR line is dropped.
- After a 1second delay, W24 raises the DTR.
- If the modem responds to the DTR drop with a **No Carrier** then Done. Otherwise, W24 issues a (+++) to the modem followed by **ATH**.

Modem Startup

Following power-up and baud rate determination, W24 in dial-up mode issues the AT<CR> command to the modem to configure the modem's baud rate.

Analog-to-Digital Converter

W24 contains an Analog-to-Digital (A/D) 8-bit converter that receives analog input voltage through the ADC signal. This input voltage can be monitored: if it reaches a predefined upper threshold or goes below a certain lower threshold, an acknowledgement can be sent. This acknowledgement is sent to the host processor through one of W24's general-purpose I/O pins (GPIO).

Input voltage can be polled every predefined number of milliseconds. In addition, a report can be obtained at any given time by issuing the AT+iRP19 command.

The following parameters determine the behavior of the A/D converter:

- ADCL and ADCD specify threshold and delta values, respectively. If the value read from the register of the A/D converter is greater than the sum of ADCL and ADCD, then the GPIO pin specified by the ADCP parameter is asserted High. If that value is less than ADCL minus ADCD, the GPIO pin is asserted Low.
- The ADCT parameter defines an interval, in milliseconds, between consecutive queries of the value of the A/D converter's register. W24's response time to value changes is up to 40ms.

In order to enable the A/D converter polling mechanism, you must, at the very least, set the ADCL, ADCT, and ADCP parameters to a non-zero value.

The following table summarizes the behavior of the A/D converter.

ADC Register Value	GPIO Pin State
$R > L+D$	High
$R < L-D$	Low

Legend:

- R - ADC register value, which is a binary representation of the A/D converter's analog input voltage.
- L - Base level, or threshold, as defined by the ADCL parameter.
- D - Delta, as defined by the ADCD parameter.

W24 Readiness Indication

This W24 Readiness Indication feature provides an indication of W24's readiness to accept AT+i commands following a hardware reset. Using this feature, W24 can also notify the host when it is ready for IP communication.

This functionality is based on two parameters - RRA and RRHW. The RRA parameter can be set to send a software message to the host, assert a dedicated hardware pin, or do both. The RRHW parameter specifies which of W24's I/O pins will be asserted.

The hardware pin specified by the RRHW parameter is asserted High immediately after power up. It will be asserted Low when W24 is ready to receive AT+i commands, and asserted High again following W24's response to any AT+i command.

Programming W24's Serial Number Into the Flash Memory

You can use the AT+iSNUM command to program the W24 serial number into the flash memory. This can be done only once.

Syntax: AT+iSNUM=<serial_number>

Programs W24's serial number into flash memory.

Parameters:

<serial_number> W24's serial number consisting of 8 hexadecimal characters. The serial number can be assigned only once, while the current serial number is still FFFFFFFF. Once a serial number is assigned, it cannot be modified. To find out the current serial number, use the AT+iRP5 command.

Default: The serial number assigned at the factory.

Result Code:

I/OK If *serial_number* is a legal hexadecimal string and is being set for the first time.

I/ERROR(068) Serial number already exists.

AT+iSNUM=? Returns the message "String" followed by I/OK.

$$AT+i\langle cc\rangle[\langle del\rangle[\langle parameter\rangle\mid\#UFn]\dots]\langle CR\rangle$$

Parameter	Description
<cc> (or <par>)	2-4 letter command code (<cc>) or parameter name (<par>)
	Delimiter: '=', '~', '?', ':', ' ',
<parameter>	Optional parameter or data. If <parameter> includes a , as defined above, it must be enclosed in single (') or double (") quotes. The terminating <CR> is considered as a terminating quote as well.
#UFn	User-field macro substitution.
<CR>	Carriage Return line terminator (ASCII 13).

Table 1-1 gives a description of AT+i commands by categories.

Command	Function	Parameters/Description
AT+i	Command prefix	Required to precede all commands.
Host Interface		
En	Echo Mode	n=0 Do not echo host characters. n=1 Echo all host characters (default upon power-up).
Parameter Database Maintenance		
<par>=value -or- <par>:value	Set parameter	<i>value</i> stored in parameter <par> in nonvolatile memory. <par> retains set value indefinitely after power down.
<par>~value	Assign single session parameter value	<i>value</i> is assigned to parameter <par> for the duration of a single Internet session. Following the session, the original value is restored.
<par>?	Read parameter	Parameter value is returned.
<par>=?	Parameter allowed values	Returns the allowed values for this parameter.
FD	Factory Defaults	Restores all parameters to factory defaults.
Status Report		
RP<i>	Request status report	Returns a status report value based on <i>.

Table 1-1: AT+i Commands by Category (Cont.)

Command	Function	Parameters/Description
Connection		
BDRA	Auto baud rate mode	Forces W24 into auto baud rate detection mode.
UP	Connect to Internet	Forces W24 to go online, establish an Internet session, and optionally register its IP address.
TUP	Triggered Internet session mode	Enters a mode in which W24 goes online in response to triggers from external signals. It also supports a special Always Online mode.
DOWN	Perform a software reset	Performs a software reset. Forces W24 to terminate an Internet session and go offline.
PING	PING a remote system	Sends a PING message and waits for its echo response.
Send E-mail		
[!]EMA:<text>	Send textual e-mail	Defines the textual contents of the e-mail body. Following this command, several text lines can be sent in sequence.
[!]EMB:<sz>, <data>	Send binary e-mail	Prefixes a binary data stream. The data is encapsulated as a base 64 encoded MIME attachment. Following this prefix, exactly <sz> bytes are streamed to W24.
[!]E*	Terminate binary e-mail	Terminates a binary (MIME attachment) e-mail.
Retrieve E-mail		
[!]RML	Retrieve mail list	Retrieves an indexed, short form list of all qualifying messages in mailbox.
[!]RMH[:<i>]	Retrieve header	Retrieves only the e-mail header part from the <i>'th e-mail in the mailbox, or the entire mailbox.
[!]RMM[:<i>]	Retrieve e-mail	Retrieves all e-mail contents of the <i>'th e-mail in the mailbox, or the entire mailbox.
HTTP Client		
[!]RLNK[:<URL>]	Retrieve link	Retrieves a file from a URL on a web server. If <URL> is not specified, uses the URL stored in the URL parameter.
[!]SLNK:<text>	Send POST request	Sends a file consisting lines of ASCII to a web server defined in the URL parameter.
HTTP Server		
WWW	Activate the web server	Activates W24's internal web server. Once activated, remote browsers can surf W24's website.
WNXT	Retrieve next changed web parameter	Returns the parameter tag name and new value of the next web parameter that has been changed as a result of a submit by a remote browser.
SerialNET		
[!]@]SNMD	Activate SerialNET mode	Activates W24's dedicated serial-to-network SerialNET mode.

Table 1-1: AT+i Commands by Category (Cont.)

Command	Function	Parameters/Description
Telnet Client		
TOPN	Telnet open session	Opens a Telnet session to a remote Telnet server. If W24 is not online, it is connected.
TRCV	Telnet receive	Receives data from a remote Telnet server.
TSND	Telnet send line	Sends an ASCII data line to a remote Telnet server.
TBSN[%]	Telnet send binary stream	Sends a binary data stream to a remote Telnet server.
TFSH[%]	Telnet flush	Flushes a Telnet socket's outbound data.
TCLS	Telnet close	Closes a Telnet session.
File Transfer Protocol (FTP)		
FOPN	Open FTP link	Opens an FTP command socket to a remote FTP server. If W24 is not online, it is connected. Once an FTP link is established, it can be used to carry out operations on the server's file system.
FOPS	Open secure FTP link	Opens an FTP link and negotiates an SSL3/TLS1 connection on the control channel. All following FTP operations in this session are performed over an SSL3/TLS1 connection.
FDL	FTP directory listing	Retrieves the remote FTP server's file directory listing. The full server-dependent listing is returned.
FDNL	FTP directory name list	Retrieves the remote FTP server's file directory listing. Only file names are returned.
FMKD	FTP make directory	Creates a directory on a remote FTP server.
FCWD	FTP change directory	Changes a remote FTP server's current directory.
FSZ	FTP file size	Retrieves the size of a file stored on a remote FTP server.
FRCV	FTP file receive	Downloads a file from a remote FTP server.
FSTO	FTP file store	Opens a file for upload to a remote FTP server. If the file already exists, it is overwritten.
FAPN	FTP file append	Opens a file on a remote FTP server for appending. If the file does not already exist, it is created.
FSND	FTP file send	Sends data to a file on a remote FTP server. The file must be already open by a previous FSTO or FAPN command.
FCLF	FTP close file	Closes the currently open file on an FTP server. Any data uploaded to the file with the FSND command is retained on the server.
FDEL	FTP delete file	Deletes a file from a remote FTP server's file system.
FCLS	FTP close	Closes a FTP link.

Table 1-1: AT+i Commands by Category (Cont.)

Command	Function	Parameters/Description
Socket Interface		
STCP: <i><host></i> , <i><port></i> [, <i><lport></i>]	Socket TCP	Opens and connects a TCP socket. If W24 is not online, it is connected. The responding system is assumed to be a server listening on the specified socket. Returns a handle to the socket.
SUDP: <i><host></i> , <i><rport></i> [, <i><lport></i>]	Socket UDP	Opens, connects, and optionally binds a UDP socket. If W24 is not online, it is connected. Returns a handle to the socket.
LTCP: <i><port></i> , <i><backlog></i>	Listening socket	Opens a TCP listening socket on <i><port></i> . Allows a maximum of <i><backlog></i> concurrent connections. Returns a handle to the socket. Up to two listening sockets are supported.
LSST: <i><hn></i>	Listening socket status	Returns a list of active socket handles accepted for a listening socket identified by handle <i><hn></i> .
SST: <i><hn></i>	Single socket status	Returns status of a single socket identified by handle <i><hn></i> . A subset of RP4 report.
SCS: <i><hn></i>	Socket connection status	Returns status of a single socket identified by handle <i><hn></i> . A subset of RP4 report. Does not report number of buffered characters.
SSND [%]: <i><hn></i> , <i><sz></i> : <i><stream></i>	Socket send	Sends a byte stream of size <i><sz></i> to the socket identified by handle <i><hn></i> . The % flag indicates automatic socket flush.
SRCV: <i><hn></i> [, <i><max></i>]	Socket receive	Receives a byte stream from the socket identified by handle <i><hn></i> . Accepts up to <i><max></i> bytes. If <i><max></i> is not specified, all available bytes are retrieved.
GPNM: <i><hn></i>	Get peer name	Retrieves peer name (<i><IP></i> : <i><port></i>) of a remote connection to the TCP/UDP socket specified by socket handle <i><hn></i> .
SDMP: <i><hn></i>	Dump socket buffer	Dumps all buffered data currently accumulated in a socket's input buffer. The socket remains open.
SFSH [%]: <i><hn></i>	Flush socket's outbound data	Flushes (sends immediately) data accumulated in a socket's outbound buffer. If the flush-and-acknowledge flag (!) is specified, W24 waits for peer to acknowledge receipt of the TCP packet.
[!]SCLS: <i><hn></i>	Close socket	Closes a TCP/UDP socket. If that socket is the only socket open and the stay online flag (!) is not specified, W24 terminates the Internet session and goes offline.
SSL: <i><hn></i>	SSL3/TLS1 socket connection	Negotiates an SSL3/TLS1 connection over an active TCP socket.
Special Modem Command		
MCM	Interlaced modem command	Sends an interlaced AT command to the modem while it is online.

Table 1-1: AT+i Commands by Category (Cont.)

Command	Function	Parameters/Description
Wireless LAN		
WLTR	WLAN transmission rate	Sets the maximum allowable WLAN transmission rate.
WLPW	WLAN Tx power	Sets the transmission power of the Marvell WLAN chipset.
WRFU	WLAN radio up	Turns on radio transmission of the Marvell WLAN chipset.
WRFD	WLAN radio down	Turns off radio transmission of the Marvell WLAN chipset.
WRST	Reset WLAN chipset	Performs a hardware reset of the Marvell WLAN chipset.
WLBM	WLAN b mode	Sets the Marvell WLAN chipset to 802.11/b mode.
WLGM	WLAN g mode	Sets the Marvell WLAN chipset to 802.11/g mode.
Remote Firmware Update		
RFU	Remote firmware update	Updates firmware from a remote HTTP or FTP server.

AT+i Result Code Summary

Table 1-2 gives the AT+i result code summary.

Table 1-2: AT+i Result Code Summary

Response String		Denotation		
I/OK		Command was successfully executed.		
I/BUSY		W24 busy. Command discarded.		
I/DONE		W24 completed Internet activity; returned to command mode, or entered SerialNET mode.		
I/ONLINE		W24 completed Internet activity and returned to command mode, or entered SerialNET mode. W24 issues this response when it has remained online as a result of the stay online flag (!) or as a result of the web server being online.		
I/OFFLINE		W24 in WLAN mode entered SerialNET Always Online mode but failed to detect a WLAN link at time of entry.		
I/RCV		Marks beginning of e-mail retrieve mode, with XFH=1. W24 does not respond to any commands, except for (+++) (Break).		
I/PART		Marks beginning of MIME attachment part.		
I/EOP		Marks end of MIME attachment part.		
I/EOM		Marks end of e-mail message during retrieve.		
I/MBE		This flag is returned when attempting to retrieve mail from an empty mailbox.		
I/UPDATE		W24 is downloading a new firmware version. Allow up to 5 minutes to complete.		
I/ERROR(nnn)	<i>nnn</i>	Command error encountered. Command discarded.		
	41	<i>Illegal delimiter</i>	42	<i>Illegal value</i>
	43	<i>CR expected</i>	44	<i>Number expected</i>
	45	<i>CR or ',' expected</i>	46	<i>DNS expected</i>
	47	<i>'.' or '~' expected</i>	48	<i>String expected</i>
	49	<i>'.' or '=' expected</i>	50	<i>Text expected</i>
	51	<i>Syntax error</i>	52	<i>',' expected</i>
	53	<i>Illegal command code</i>	54	<i>Error when setting parameter</i>
	55	<i>Error when getting parameter value</i>	56	<i>User abort</i>
	57	<i>Error when trying to establish PPP</i>	58	<i>Error when trying to establish SMTP</i>
	59	<i>Error when trying to establish POP3</i>	60	<i>Single session body for MIME exceeds the maximum allowed</i>
	61	<i>Internal memory failure</i>	62	<i>User aborted the system</i>

Table 1-2: AT+i Result Code Summary (Cont.)

Response String		Denotation		
	63	<i>~CTSH needs to be LOW to change to hardware flow control.</i>	64	<i>User aborted last command using '---'</i>
	65	<i>RESERVED</i>	66	<i>RESERVED</i>
	67	<i>Command ignored as irrelevant</i>	68	<i>W24 serial number already exists</i>
	69	<i>Timeout on host communication</i>	70	<i>Modem failed to respond</i>
	71	<i>No dial tone response</i>	72	<i>No carrier modem response</i>
	73	<i>Dial failed</i>	74	<i>Modem connection with ISP lost -or- WLAN connection lost</i>
	75	<i>Access denied to ISP server</i>	76	<i>Unable to locate POP3 server</i>
	77	<i>POP3 server timed out</i>	78	<i>Access denied to POP3 server</i>
	79	<i>POP3 failed</i>	80	<i>No suitable message in mailbox</i>
	81	<i>Unable to locate SMTP server</i>	82	<i>SMTP server timed out</i>
	83	<i>SMTP failed</i>	84	<i>RESERVED</i>
	85	<i>RESERVED</i>	86	<i>Writing to internal non-volatile parameters database failed</i>
	87	<i>Web server IP registration failed</i>	88	<i>Socket IP registration failed</i>
	89	<i>E-mail IP registration failed</i>	90	<i>IP registration failed for all methods specified</i>
	91	<i>RESERVED</i>	92	<i>RESERVED</i>
	93	<i>RESERVED</i>	94	<i>In Always Online mode, connection was lost and re-established</i>
			96	<i>A remote host, which had taken over W24 through the LATI port, was disconnected</i>
			98	<i>RESERVED</i>
	99	<i>RESERVED</i>	100	<i>Error restoring default parameters</i>
	101	<i>No ISP access numbers defined</i>	102	<i>No USRN defined</i>
	103	<i>No PWD entered</i>	104	<i>No DNS defined</i>
	105	<i>POP3 server not defined</i>	106	<i>MBX (mailbox) not defined</i>
	107	<i>MPWD (mailbox password) not defined</i>	108	<i>TOA (addressee) not defined</i>
	109	<i>REA (return e-mail address) not defined</i>	110	<i>SMTP server not defined</i>

Table 1-2: AT+i Result Code Summary (Cont.)

Response String		Denotation		
	111	Serial data overflow	112	Illegal command when modem online
	113	E-mail firmware update attempted but not completed. The original firmware remained intact	114	E-mail parameters update rejected
	115	SerialNET could not be started due to missing parameters	116	Error parsing a new trusted CA certificate
	117	RESERVED	118	Protocol specified in the USRV parameter does not exist or is unknown
	119	WPA passphrase too short - has to be 8-63 chars	120	RESERVED
	121	RESERVED	122	SerialNET error: Host Interface undefined (HIF=0)
	123	SerialNET mode error: Host baud rate cannot be determined	124	SerialNET over TELNET error: HIF parameter must be set to 1 or 2
	125	Invalid WEP Key	200	Socket does not exist
	201	Socket empty on receive	202	Socket not in use
	203	Socket down	204	No available sockets
			206	PPP open failed for socket
	207	Error creating socket	208	Socket send error
	209	Socket receive error	210	PPP down for socket
			212	Socket flush error
	215	No carrier error on socket operation	216	General exception
	217	Out of memory	218	An STCP (Open Socket) command specified a local port number that is already in use
	219	SSL initialization/internal CA certificate loading error	220	SSL3 negotiation error
	221	Illegal SSL socket handle. Must be an open and active TCP socket.	222	Trusted CA certificate does not exist
	223	RESERVED	224	Decoding error on incoming SSL data
	225	No additional SSL sockets available	226	Maximum SSL packet size (2K) exceeded
	227	AT+iSSND command failed because size of stream sent exceeded 2048 bytes	228	AT+iSSND command failed because checksum calculated does not match checksum sent by host

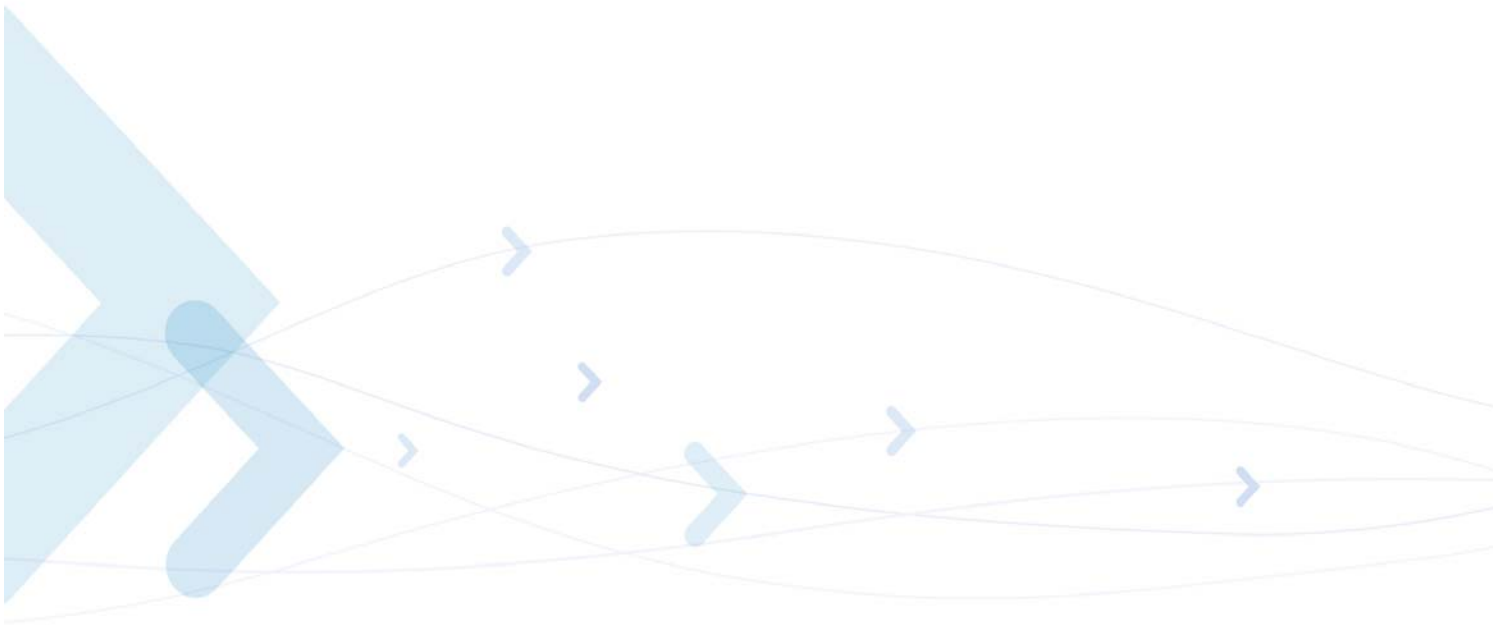
Table 1-2: AT+i Result Code Summary (Cont.)

Response String		Denotation		
			300	<i>HTTP server unknown</i>
	301	<i>HTTP server timeout</i>	302	<i>HTTP failure</i>
	303	<i>No URL specified</i>	304	<i>Illegal HTTP host name</i>
	305	<i>Illegal HTTP port number</i>	306	<i>Illegal URL address</i>
	307	<i>URL address too long</i>	308	<i>The AT+iWWW command failed because W24 does not contain a home page</i>
			400	<i>MAC address exists</i>
	401	<i>No IP address</i>	402	<i>Wireless LAN power set failed</i>
	403	<i>Wireless LAN radio control failed</i>	404	<i>Wireless LAN reset failed</i>
	405	<i>Wireless LAN hardware setup failed</i>	406	<i>Command failed because WiFi module is currently busy</i>
	407	<i>Illegal WiFi channel</i>	408	<i>Illegal SNR threshold</i>
			500	<i>RESERVED</i>
	501	<i>Communications platform already active</i>	502	<i>RESERVED</i>
	503	<i>RESERVED</i>	504	<i>RESERVED</i>
	505	<i>Cannot open additional FTP session - all FTP handles in use</i>	506	<i>Not an FTP session handle</i>
	507	<i>FTP server not found</i>	508	<i>Timeout when connecting to FTP server</i>
	509	<i>Failed to login to FTP server (bad username or password or account)</i>	510	<i>FTP command could not be completed</i>
	511	<i>FTP data socket could not be opened</i>	512	<i>Failed to send data on FTP data socket</i>
	513	<i>FTP shutdown by remote server</i>	514	<i>RESERVED</i>
			550	<i>Telnet server not found</i>
	551	<i>Timeout when connecting to Telnet server</i>	552	<i>Telnet command could not be completed</i>
	553	<i>Telnet session shutdown by remote server</i>	554	<i>A Telnet session is not currently active</i>
	555	<i>A Telnet session is already open</i>	556	<i>Telnet server refused to switch to BINARY mode</i>
	557	<i>Telnet server refused to switch to ASCII mode</i>	558	<i>RESERVED</i>
	559	<i>RESERVED</i>	560	<i>Client could not retrieve a ring response e-mail</i>

Table 1-2: AT+i Result Code Summary (Cont.)

Response String		Denotation		
	561	<i>Remote peer closed the SerialNET socket</i>		
			570	<i>PING destination not found</i>
	571	<i>No reply to PING request</i>		

Note: All W24 response strings are terminated with <CR><LF>.



Chapter 2: AT+i Commands Reference

Report Status

+i[!]RPi - Report Status

Syntax: AT+i[!]RPi

Returns a status report.

Parameters: i=0..20

Command Options:

i=0 Returns the W24 part number.

i=1 Returns the current firmware revision and date.

i=2 Returns the connection status.

i=3 Returns boot-block revision and date.

i=4 Returns W24 socket status.

i=5 Returns a unique serial number.

i=6 Returns current ARP table.

i=7 Returns socket buffers utilization bitmap. W24's DATA_RDY signal can be used to signal socket buffer status changes in hardware. This signal is raised when new data in one or more sockets is available, or when a remote browser has changed a web parameter. It is lowered when **any** socket or web parameter is read.

i=8 Returns current time-of-day based on time retrieved from the Network Time Server and the GMT offset setting. Returns an all-zero response if a timestamp has not yet been retrieved from the network since the last power-up.

i=9 Reserved

i=10 Return two different status reports about the current Wireless LAN connection.

AT+i!RP10

i=11 Returns a list of all Access Points available in the surrounding area.

AT+!RP11	Returns a list of all ad-hoc networks available in the surrounding area.
<i>i</i> =14	Returns a DHCP server table of MAC and IP addresses of all the stations connected to W24.
<i>i</i> =19	Returns Analog-to-Digital Converter (ADC) pin status report.
<i>i</i> =20	Returns a list of all APs and ad-hoc networks available in the surrounding area.
Default:	None
Result Code:	
<i>i</i> =0..20	Status message followed by I/OK .
I/ERROR	Otherwise

Status Message Format

Table 2-1 gives the report status message format.

Table 2-1: Report Status Message Format

Report Option	Format		
0	COnnnAD- <i>ii</i> <i>nnn</i> – Version number; <i>ii</i> – Interface code: S-Serial, L-LAN, D-Dual		
1	IiimmmTss (<version-date>) <i>Iii</i> – Interface code; <i>mmm</i> – Major Version; <i>T</i> – Version type code; <i>ss</i> – Sub-version		
2	Status string: "Modem data<CR/LF>" "Command mode<CR/LF>" "<CR/LF>Connecting to ISP<CR/LF>" "<CR/LF>Connected to ISP<CR/LF>" "<CR/LF>Connecting as RAS<CR/LF>" "<CR/LF>RAS Connected<CR/LF>""<CR/LF>Closing PPP<CR/LF>" "<CR/LF>Establishing SMTP<CR/LF>" "<CR/LF>Sending Email<CR/LF>" "<CR/LF>Establishing POP3<CR/LF>" "<CR/LF>POP3 Open<CR/LF>" "<CR/LF>Establishing HTTP<CR/LF>" "<CR/LF>Receiving HTTP<CR/LF>" ""<CR/LF>Carrier Lost<CR/LF>" ""<CR/LF>Link Lost<CR/LF>"		
3	<i>nnmm</i> – Boot block version number		
4	I/(<sock0sz>, <sock1sz>, ... ,<sock9sz>) <i>sock<i>sz</i> >=0	: Number of bytes pending in socket's input buffer <0	: Negative value of socket's error code
5	<i>nnnnnnnn</i> – Hexadecimal representation of W24 serial number		

Table 2-1: Report Status Message Format (Cont.)

Report Option	Format																																		
6	Current ARP table listing: INTERNET ADDRESS PHYSICAL ADDRESS STATE TTL nnn.nnn.nnn.nnn xxxxxxxxxxxxxx VALID nnn sec. For debugging purposes.																																		
7	I/xxxx xxxx – 16 bit Hex Value Bitmap A bit set to ‘1’ indicates that the corresponding socket contains buffered data, which needs to be read by the host. <table><tr><td>bit</td><td>15</td><td></td><td></td><td></td><td></td><td>10</td><td></td><td></td><td>7</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0</td></tr><tr><td>socket</td><td></td><td></td><td></td><td></td><td></td><td>WEB</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr></table> Bit 10 is set to ‘1’, when the remote browser updates <i>one or more</i> application website parameter tags. It will be reset to ‘0’ when the host reads <i>any</i> application website parameter, using AT+i<Parameter Tag>?	bit	15					10			7							0	socket						WEB	9	8	7	6	5	4	3	2	1	0
bit	15					10			7							0																			
socket						WEB	9	8	7	6	5	4	3	2	1	0																			
8	The current time-of-day is returned according to ISO 8601: <YYYY-MM-DD>T<HH:MM:SS> <TZD> YYYY-MM-DD -- Year-Month-Day ; ‘T’ – Fixed Separator ; HH:MM:SS - Hrs:Mins:Secs ; TZD - Time Zone Designator: +hh:mm or –hh:mm All-zeros response: 0000-00-00T00:00:00 <TZD>.																																		
9	Reserved																																		
10	I/(<port stat>, <xfer rate>, <sig level>, <lnk qual>) port stat-Port Status:0: Wireless LAN adapter not present 1: Wireless LAN adapter disabled 2: Searching for initial connection 4: Connected 5: Out of range xfer rate-- Transfer rate in the range 1..54 sig level-- Signal level [%], in the range 0..100 lnk qual-- Link quality [%], in the range 0..100 I/OK																																		
AT+i!RP10	Returns a report of the current WLAN connection. <SSID>,<BSSID>,<security type>,<WPA status>,<channel>,<SNR> I/OK where ▪ <security type>=NONE WEP64 WEP128 WPA WPA2 ▪ <WPA status>=Completed Not Completed <WPA status> indicates, when WPA/WPA2 security is specified, whether WPA negotiation completed or not. Notes: ▪ For ad-hoc networks, SSID starts with (!), if it is the ad-hoc creator. ▪ WPA status is reported whether WPA negotiation completed or not. For example: Jetta,06:14:6C:69:4A:7C,WPA,Completed,1,68 I/OK																																		

Table 2-1: Report Status Message Format (Cont.)

Report Option	Format
11	<p>W24 scans all available Access Points (APs) in the surrounding area and returns a list of APs. Each line contains the following comma-separated fields: SSID, security scheme, and signal strength. The AP having the strongest signal appears first.</p> <pre><SSID>,<security_scheme>,<signal_strength><CR><LF> <SSID>,<security_scheme>,<signal_strength><CR><LF> . . <SSID>,<security_scheme>,<signal_strength><CR><LF> I/OK<CR><LF></pre> <p>where,</p> <p><i>SSID</i> – Up to 32 alphanumeric characters <i>security_scheme</i> –None WEP WPA <i>signal_strength</i>0 - low, 1 - good, 2 - excellent</p> <p>Note: If no APs are detected, only I/OK<CR><LF> is returned.</p>
AT+i!RP11	<p>Returns a list of all ad-hoc networks available in the surrounding area. Each line contains the following comma-separated fields: SSID, security scheme, and signal strength. Security scheme for ad-hoc networks is NONE. The ad-hoc network having the strongest signal appears first.</p> <pre><SSID>,<NONE>,<signal_strength><CR><LF> <SSID>,<NONE>,<signal_strength><CR><LF> . . <SSID>,<NONE>,<signal_strength><CR><LF> I/OK<CR><LF></pre> <p>where</p> <p><i>SSID</i> – Up to 32 alphanumeric characters <i>signal_strength</i>0 - low, 1 - good, 2 - excellent</p> <p>For example: Free Public WiFi,NONE,1 I/OK<CR><LF></p> <p>Note: If no ad-hoc networks are detected, only I/OK<CR><LF> is returned.</p>
14	<p>Returns a DHCP server table of MAC and IP addresses of all the stations connected to W24.</p> <pre>MAC AddressIP Address <MAC_Address_1><IP_Address_1> . <MAC_Address_n><IP_Address_n> I/OK</pre> <p>For example: MAC AddressIP Address 00039406068C192.168.0.2 000394094D1B192.168.0.3 I/OK</p>

Table 2-1: Report Status Message Format (Cont.)

Report Option	Format
19	<p>Returns Analog-to-Digital Converter (ADC) pin status report. If the ADCP parameter is set, the reports returns GPIO pin state. Otherwise, it returns the ADC value only. ADC value=<level>, GPIO state=<state> I/OK where</p> <ul style="list-style-type: none"> • <i>level</i> is an integer in the range 0-255 representing the input voltage measured on the ADC pin, calculated as follows: $(A/3.3V)*255=level$, where A is the analog input voltage. • <i>state</i> indicates the state of the output GPIO pin: 0 (High) 1 (Low). GPIO state is reported only if the ADCL, ADCT and ADCP parameters are set. <p>For example, if the ADCP parameter is set: ADC value = 255, GPIO state = 0 I/OK If the ADCP parameter is not set: ADC value = 255 I/OK</p>
20	<p>Returns a list of all APs and ad-hoc networks available in the surrounding area. Each line contains the following comma-separated fields: <SSID>,<ADHOC AP,<BSSID>,<securitytype>,<channel>,<RSSI> I/OK where <security type>=NONE WEP WPA WPA2 <RSSI>= SNR+NoiseFloor For example: Jetta,AP,06:14:6C:69:4A:7C,WPA,1,25 RTL8186-default,AP,00:E0:4C:81:86:86,NONE,1,77 dlink_test,AP,00:1C:F0:9A:63:7A,NONE,1,68 Guest,AP,00:15:E9:0C:38:F2,WPA2,6,69 ABC,AP,00:1C:F0:40:CC:60,NONE,6,65 Yuval,AP,00:0E:2E:C6:B6:E1,NONE,6,62 GANG_TEST,AP,00:17:3F:9F:89:6E,NONE,7,67 Bora,AP,00:14:78:F7:11:BA,NONE,7,26 3com_test,AP,00:0F:CB:FF:27:8F,NONE,7,81 INET,AP,00:0F:CB:FF:7E:5D,WPA,7,82 Blue-I The Lab,AP,00:1B:2F:57:65:62,WEP,7,45 Mistral,AP,00:11:6B:3B:55:E2,WEP,9,27 Sirocco,AP,00:18:4D:DE:D7:DF,WPA2,11,44 Free Public WiFi,ADHOC,D2:B3:5B:06:CA:04,NONE,11,69 BlueI,AP,00:0E:2E:55:39:A6,WEP,11,57 private,AP,00:0E:2E:FD:F0:69,WPA,11,74 I/OK</p>

Connection

+iBDRA - Force W24 into Auto Baud Rate Mode

Syntax: AT+iBDRA

Forces the W24 into auto baud rate mode. The following A, AT or AT+i command (in any combination of upper or lowercase) from the host will synchronize on the host's baud rate. W24 supports auto baud rate detection for the following baud rates: 2400, 4800, 9600, 19200, 38400, 57600, and 115200.

Result Code:

I/OK This result code is sent using the previous baud rate.

+iUP - Initiate Internet Session

Syntax: AT+iUP[:*n*]

Initiates an Internet session by going online. In a dialup/cellular environment, a PPP Internet connection is established. Once online, optionally goes through an IP registration process, as determined by *n*.

Parameters: *n*=0..1

Default: *n*=0

Command Options:

n=0 Go online.

n=1 Go online and carry out the IP registration process according to the relevant registration option parameters.

Result Code:

I/ONLINE After successfully establishing an Internet session and completing the IP registration (if requested).

I/ERROR If W24 cannot go online and establish an Internet session or cannot complete the requested IP registration.

+iTUP - Triggered Internet Session Initiation

Syntax: AT+iTUP:<*n*>

Enter triggered Internet session initiation mode.
This command is relevant in a modem environment only.

Parameters: *n*=0..2

Command Options:

n=0 Disable triggered Internet session initiation mode.

n=1 Enter triggered Internet session initiation mode. Upon receiving a hardware signal trigger (Modem RING or MDSEL signal pulled low), establish a PPP Internet connection and carry out the IP registration process according to the relevant registration option parameters.

If any characters are received on the host port prior to receiving a hardware signal, W24 exits this mode and functions normally. In this case, to reinstate this mode, issue AT+iTUP=1 again; reset W24 by issuing the AT+iDOWN command, or recycle power.

n=2 Always Online mode. Whenever W24 is offline, it automatically attempts to establish a PPP Internet connection and possibly carry out the IP registration process according to the relevant registration option parameters.

W24 disregards this mode and remains offline until the next SW or HW reset if:

- The MSEL (Mode Select) signal was pulled low (logical 0) for more than 5 seconds during runtime.

-or-

- The host issues the (+++) escape sequence.

Power must be recycled or the AT+iDOWN command issued for this command to take effect.

If W24 is in Auto Baud Rate mode (BDRF=a) and/or Auto Host mode (HIF=0), W24 waits for the a character on the host serial port to resolve the baud rate after rebooting and before activating the iRouter and going online, or before activating the DHCP server. Therefore, it is recommended to set a fixed host interface and a fixed baud rate in this case.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

Notes:

- When going online in one of these modes, W24 activates its web server if the AWS parameter is set (AWS>0).
- In this mode, W24 does not go offline after a completion of any successful or unsuccessful Internet session started by the host, even if the stay online flag is not used.
- When a Carrier Lost event is detected, W24 automatically retries to establish a connection (without performing a software reset), with the following exception: If, at the time of the detection, the host was waiting for a reply from W24 or was in the process of sending binary data (SSND, FSND, EMB), W24 reports error code 094 as soon as it can and only then tries to re-establish the connection. In all other cases, W24 gives the host no indication of losing the carrier. In the event of Carrier Lost, W24 closes any

open TCP active sockets, but leaves UDP sockets and TCP passive (listening) sockets intact and updates their local IP if a new IP is assigned after establishing a new PPP connection. W24 does not close any open Internet sessions (FTP/Telnet sessions and so on), nor releases the handle of the active TCP sockets, thus giving the host a chance to read the session errors and get buffered incoming data from active TCP sockets.

- When the PFR is larger than 0 and the PDSn parameters are configured, W24 verifies that it is online by sending PING messages to the PING destination servers defined in PDSn at a polling frequency defined by PFR. If both PING destination servers do not respond, W24 concludes that the Internet connection failed and tries to reestablish an Internet connection, as described above for the case of a lost carrier signal.

+iDOWN - Terminate Internet Session

Syntax: AT+iDOWN

Performs a software reset. Terminates an ongoing Internet session, goes offline and returns to Command mode.

This command is useful in a dialup environment following a command where the stay online flag (!) was specified.

All open sockets are closed and the web server deactivated.

Result Code:

I/OK If *n* is within limits.

Followed by:

I/ERROR After terminating the current Internet session when the command caused W24 to abort an ongoing Internet activity or close an active socket.

-or-

I/DONE After terminating the current Internet session. Allow a 2.5 sec. delay for W24 re-initialization following an Internet mode session. Relevant for W24 in dial-up mode only.

-or-

I/ONLINE After terminating the current Internet session.

+iPING - Send a PING Request to a Remote Server

Syntax: AT+iPING:<host>

Sends a two-byte ICMP PING request packet to the remote host defined by *host*.

Parameters: <host>=Logical name of the target host or a host IP address.

Command Options:

<host> The host name may be any legal Internet server name, which can be resolved by the W24's DNS (Domain Name Server) settings. The host name may also be specified as an absolute IP address given in DOT form.

Result Code:

I/<RTT> Upon successfully receiving an ICMP PING reply from the host, the round trip time in milliseconds is returned (RTT). W24 allows up to <PGT> milliseconds for a PING reply. If a reply is not received within <PGT> milliseconds, W24 sends two more PING requests, allowing <PGT> milliseconds for a reply on each of the requests before reporting an error.

I/ERROR Otherwise.

E-mail Send Commands

+iEMA - Accept ASCII-Coded Lines for E-Mail Send

Syntax: AT+i[!]EMA:<text lines>

Defines a plain text e-mail body.

Parameters:

<text lines> Plain text e-mail body. The e-mail body contains <CR/LF> terminated ASCII character strings. <text lines> must be terminated by a dot character (.) in the 1st column of an otherwise empty line.

Command Options: <text lines>::={<ASCII text line><CRLF> ...}<CRLF>.<CRLF>
Maximum size of <text lines> is limited to 18K, provided that no additional system resources are in use.
EMA uses the specified SMTP server to send the e-mail message. When W24 acquires TOD from a network timeserver, outgoing e-mail messages are time and date stamped.

! Stay online after completing the command.

Result Code:

I/OK After all text lines are received and terminated by the (.) line.

I/ERROR If memory overflow occurred before all text lines are received.

Followed by:

I/DONE After successfully sending the e-mail. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully sending the e-mail, if the stay online flag (!) is specified.

-or-

I/ERROR If some error occurred during the send session.

+iEMB - Accept Binary Data for Immediate E-Mail Send

Syntax: AT+i[!]EMB[#]:<sz>,<data>

Defines and sends a MIME-encoded binary e-mail.

Parameters:

<sz> Size of <data> in bytes.

<data> <sz> bytes of binary data.

Command Options:

<sz> 0..4GB

<data> 8 bit binary data. Must be exactly <sz> bytes long.
 The binary data is encapsulated in a MIME-encoded e-mail message. The receiving end views the binary data as a standard e-mail attachment. Several consecutive +iEMB commands can be issued in sequence to create a larger aggregate of data to be sent.
 The e-mail contents are completed by issuing an AT+iE* (terminate binary e-mail) command. Following the first +iEMB command, W24 establishes an Internet connection while the data stream is being transmitted from the host. Once an SMTP session is established, W24 maintains a data transmit pipeline between the host and the SMTP server. W24 converts the binary data using BASE64 encoding on-the-fly. Following this command, the Internet session remains active to service additional +iEMB commands, until the +iE* terminating command.
 EMB uses the specified SMTP server to send the e-mail message. When W24 acquires TOD from a network timeserver, outgoing e-mail messages are time and date stamped.
 ! Stay online after completing the command. This flag is redundant, as the W24 defaults to staying online until the AT+iE* command is issued.
 # Modem baud rate limit flag. When this character is included in the command, the W24 baud rate to the modem is limited by the baud rate from the host. This flag is relevant for serial modems only and is especially useful in GSM modem configurations. When this character is not present, the W24 attempts to lift the baud rate to the modem to its maximal value.

Result Code:

I/OK If <sz> is within limits and after <sz> bytes have been received successfully.

I/ERROR If <sz> is out of bounds, or if a communication error occurred during the Internet session.

Notes:

- If <sz> is larger than 256 bytes, W24 assumes host flow control. Depending on the setting of the FLW parameter, the flow control mode is either software or hardware. Under software flow control, the host processor must respond to W24's flow control characters. The software flow control protocol is detailed in the Host -> W24 Software Flow Control section later in this document. When software flow control is active, it is recommended to set the W24 to Echo-Off mode. Under hardware flow control, the ~CTS/~RTS RS232 control signals must be connected and the host must respond to the W24's ~CTS signal. The host may send data only when the ~CTS signal is asserted

(active low). If a transmission error occurs while in hardware flow control, W24 continues receiving all remaining <sz> bytes before returning the I/ERROR response.

- Some SMTP servers limit e-mail message size to a value that is lower than W24's limitations.

+iE* - Terminate Binary E-Mail

Syntax: AT+i[!]E*

Terminates the current binary e-mail attachment.

Parameters:

<text lines> Plain text e-mail body. The e-mail body contains <CR/LF> terminated ASCII character strings. <text lines> must be terminated by a dot character (.) in the 1st column of an otherwise empty line.

Command Options:

! Stay online after completing the command.

Result Code:

I/OK If a binary e-mail attachment is in the process of being defined. The e-mail message is terminated and the SMTP session is then completed and closed.

I/ERROR Otherwise.

Followed by:

I/DONE After successfully sending the e-mail. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully sending the e-mail, if the stay online flag (!) is specified.

-or-

I/ERROR If some error occurred during the send session.

E-Mail Retrieve

+iRML - Retrieve Mail List

Syntax: AT+i[!]**RML**

Retrieves pending e-mail list from current mailbox.

Command Options:

! Stay online after completing the command.

Result Code:

I/OK To acknowledge successful receipt of the command.

I/ERROR Otherwise.

Returns:

I/MBE If the mailbox is empty.

Otherwise: A list of qualifying e-mail message descriptors, separated by <CR/LF>. An e-mail message descriptor is composed of 5 <TAB> separated fields:

<i><TAB><sz><TAB><date><TAB><sbjct string>
<TAB><type/subtype><CR/LF>

where,

<i> - E-mail message index in mailbox

<sz> - E-mail message size in bytes

<date> - E-mail message date (for the date field format refer to RFC822)

<sbjct string>- E-mail message subject string (limited to 128 bytes)

<type/subtype> - MIME content type. The literal NONE is used for non-MIME e-mail messages.

E-mail messages that qualify the E-Mail Delete Filter (DELF) are not listed.

Followed by:

I/DONE After successfully sending the e-mail. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully retrieving the e-mail list, if the stay online flag (!) is specified.

I/ERROR Otherwise.

+iRMH - Retrieve Mail Header

Syntax: AT+i[!]**RMH[:i]**

Retrieves header of e-mail message <i> from current mailbox.

Parameters:

- i* Optional e-mail message index of a qualifying message. If no parameter is used, all e-mail headers are retrieved.

Command Options:

- i* Optional index of a qualifying message, as reported by AT+IRML.
! Stay online after completing the command.

Default: Retrieves headers of all pending qualified mail messages.

Result Code:

- I/OK** When command is received and about to be processed.
I/ERROR Otherwise.

Returns:

- I/MBE** If the mailbox is empty.

Otherwise: All header lines of all qualifying e-mail messages. Header lines are returned as-is. A line containing solely a (.) (period) in column 1 acts as a separator between the header lines of each e-mail. The HDL parameter limits the number of header lines per mail (HDL=0 specifies an unlimited number of lines per e-mail). Header field syntax is described in RFC822 and RFC2045.

Followed by:

- I/DONE** After successfully sending the e-mail headers. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

- I/ONLINE** After successfully retrieving the e-mail headers, if the stay online flag (!) is specified.
I/ERROR Otherwise.

+iRMM - Retrieve Mail Message

Syntax: AT+[!]*RMM*[:*i*]

Retrieves contents of e-mail message *i* from current mailbox.

Parameters:

- i* Optional e-mail message index of a qualifying message. If no parameter is used, all e-mails are retrieved.

Command Options:

- i* Optional index of a qualifying message, as reported by AT+IRML.
! Stay online after completing the command.

Default: Retrieves all pending qualified mail messages.

Result Code:

- I/OK** When command is received and about to be processed.

I/ERROR Otherwise.

Returns:

I/MBE If the mailbox is empty.

Otherwise: For each e-mail part:
(For plain-text e-mails without MIME attachments)

I/PART - *<text><TAB><plain><TAB><TAB>
<quoted-printable><CR/LF>*

-or- (For e-mails containing MIME attachments)

I/PART - *<media type><TAB><media subtype><TAB>
<filename><TAB> <encoding method><CR/LF>*

-or- (When XFH - transfer e-mail headers - is set to YES)

I/RCV

-or-

Followed by: *<e-mail message contents>*

If the XFH parameter (transfer e-mail headers) is set to YES, all e-mail contents are returned as-is. The e-mail's headers followed by the e-mail's body are retrieved. MIME encapsulated e-mail messages are retrieved without BASE64 decoding. It is assumed that when the XFH parameter is set to YES, the host processor attends to all e-mail field parsing and contents decoding.

If the XFH parameter is set to NO, only the email's body (contents) are retrieved. If the email message contains a MIME-encapsulated attachment encoded in BASE64, W24 performs the decoding and transfers pure binary data to the host. Binary attachments encoded in a scheme other than BASE64 are returned as-is.

E-mails that qualify the Delete E-Mail Filter (DELF) are deleted from the mailbox without being downloaded.

Followed by:

I/EOP End of Part Message, if message is prefixed with an **I/PART** line.

This repeats itself for all e-mail parts.

Followed by:

I/EOM End of Message

This repeats itself for all qualifying e-mail messages.

When all messages
have been retrieved:

I/DONE After successfully retrieving the e-mail. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully retrieving the e-mail, if the stay online flag (!) is specified.

I/ERROR Otherwise.

Figure 2-1 shows E-Mail Receive (RMM) flow diagram.

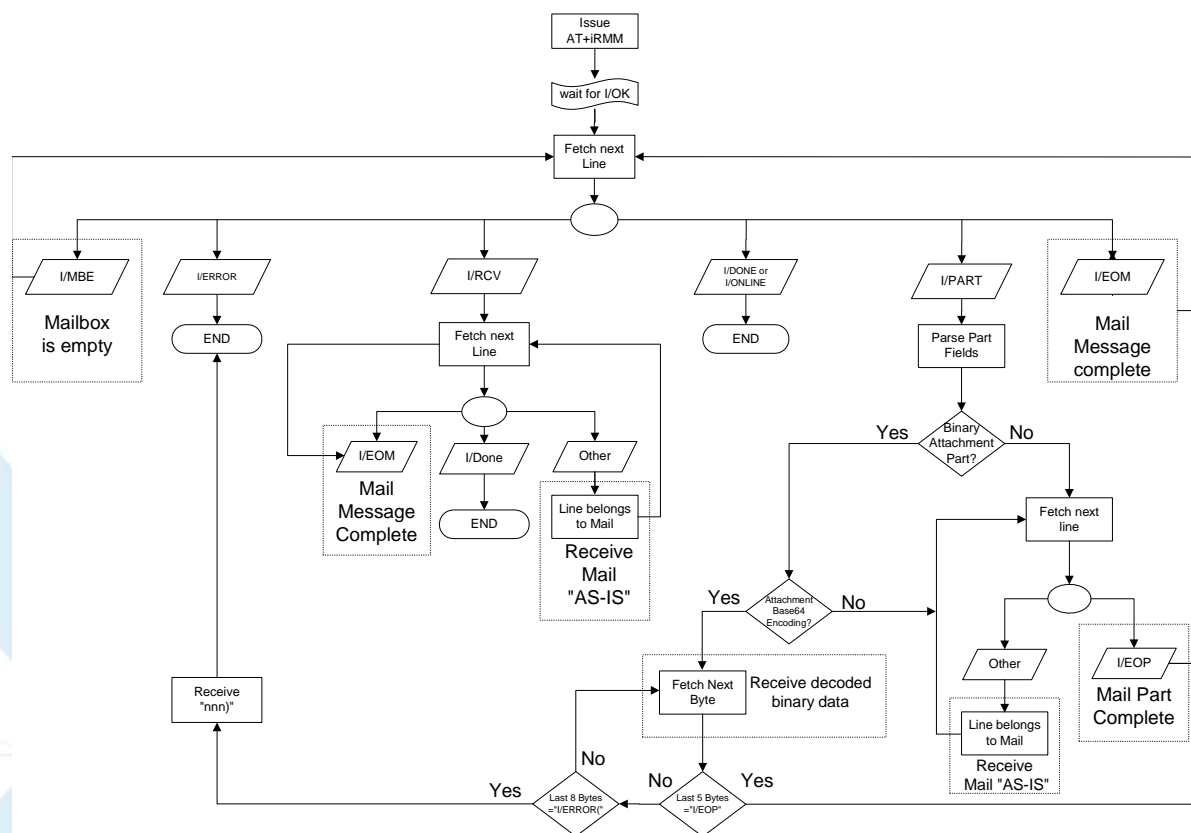


Figure 2-1: E-Mail Receive (RMM) Flow Diagram

HTTP Client Interface

+iRLNK - Retrieve Link

Syntax: AT+i[!]**RLNK**[:*URL*]

Retrieves a file from a URL.

Parameters: *URL* = Optional URL address, which specifies the host, path, and source file to be retrieved.

URL address syntax:

"<protocol>://<host>[:<port>]/[<abs_link>]/"

Command Options:

<protocol> http or https.

<host> Host name or IP address.

<port> 0..65535
If not specified, defaults to 80 for http and 443 for https.

<abs_link> Path, filename, and file extension of the file to retrieve on the designated host.

! Stay online after completing the command.

Default: Uses the URL address stored in the URL parameter.

Result Code:

I/OK When command is received and about to be processed.

I/ERROR Otherwise.

Returns: I/<sz><CR><LF>

Followed by: <binary data stream>
where,

<sz> is the exact size of the <binary data stream> to follow.

If <sz> is unknown, W24 returns **I/0** followed by the data stream. When this is the case, the host must monitor for a timeout condition of at least 5 seconds without any data being transmitted before seeing one of the terminator lines described under 'Followed by'.

Followed by:

I/DONE After successfully retrieving the file. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully retrieving the file, if the stay online flag (!) is specified.

-or-

I/ERROR Otherwise. (Always preceded by a 5 seconds silence period.)

+iSLNK - Submit a POST Request to a Web Server

Syntax: AT+i[!]SLNK:<text>

Submits a plain text POST request to a web server defined in the URL parameter. The "Content-type:" field of the POST request is defined by the CTT parameter.

Parameters: <text> = Plain text POST request body containing <CR[LF]> terminated ASCII character strings. <text> must be terminated by a dot character (.) in the first column of an otherwise empty line.

Command Options:

<text> <ASCII text line><CRLF> ...<CRLF>.<CRLF>
Maximum size of <text> depends on the amount of memory available in the specific W24. SLNK uses the URL address stored in the URL parameter to send the POST request.

! Stay online after completing the command.

Result Code:

I/OK After all text lines are received from the host.

I/ERROR If a memory overflow occurred before all text lines are received.

SerialNET Mode Initiation

+iSNMD - Activate SerialNET Mode

Syntax: AT+i[! | @]SNMD

Activates SerialNET mode. Instead of using the optional (!) and (@) flags, you can use the following syntax:

AT+iSNMD=1 is equivalent to AT+iSNMD

AT+iSNMD=2 is equivalent to AT+i!SNMD

AT+iSNMD=3 is equivalent to AT+i@SNMD

AT+iSNMD=4 causes W24 to enter SerialNET over TELNET mode

Command Options:

AT+i!SNMD
-or-
AT+iSNMD=2

Optional Auto-Link mode. When this flag is specified, W24 immediately goes online when activating SerialNET mode (even when serial data has not yet arrived). If the LPRT (Listening Port) parameter is defined, W24 opens the listening port and awaits a connection. If LPRT is not defined, but HSRV (Host Server) is defined, W24 immediately opens a SerialNET socket link to the server.

AT+i@SNMD
-or-
AT+iSNMD=3

Optional Deferred Connection mode. When this flag is specified, W24 automatically goes online (as in the case of AT+i!SNMD). However, if the HSRV parameter is defined, a socket is not opened until data arrives on the local serial port.

If the SerialNET mode listening port is defined (LPRT), W24 opens a listening socket and waits for a remote connection during the idle period before data arrives on the local serial port.

When the SerialNET socket type (STYP) is TCP and serial data arrives, W24 buffers the data in the MBTB Buffer and tries to connect to HSR0. If HSR0 does not respond, W24 tries HSR1, then HSR2. If all three connection attempts fail, W24 retries them all. After three full retry cycles, W24 dumps the MBTB buffer and remains idle until new serial data arrives.

AT+iSNMD=4

Optional SerialNET over TELNET mode. In this mode, W24 opens a data socket as a TELNET socket, which allows negotiations of TELNET options over the same socket while the host is sending and receiving raw data only. This mode partially supports the RFC2217 standard.

Before issuing this command, you must set W24's Host Interface to USART0 (HIF=1) or USART1 (HIF=2). For more information about this mode, refer to the SerialNET over TELNET description.

Result Code:

I/OK If all minimum required parameters for SerialNET mode operation are defined (HSRV or LPRT and, in a modem environment, also ISP1, USRN, PWD).

I/ERROR Otherwise.

Followed by:

I/DONE After successfully activating SerialNET mode when using. Allow a 2.5 seconds delay for W24 re-initialization.

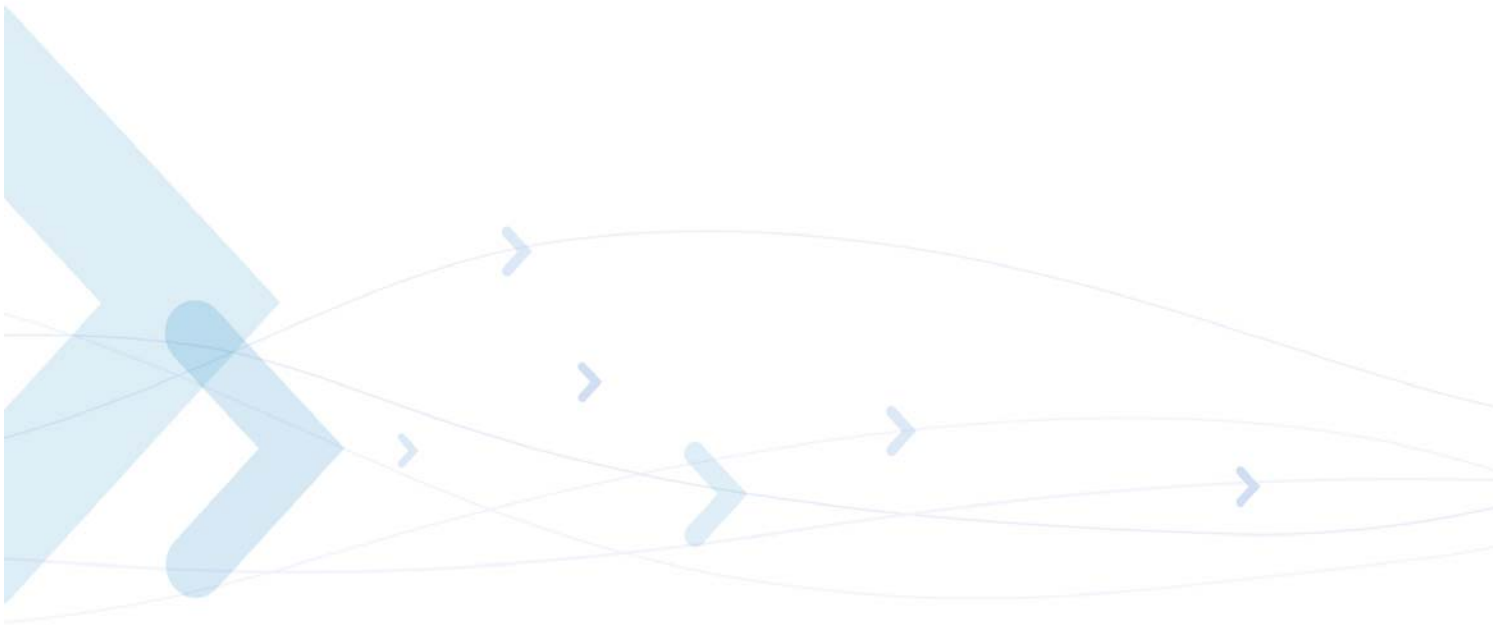
-or-

I/ONLINE After successfully activating SerialNET mode. Allow a 2.5 seconds delay for W24 re-initialization.

-or-

I/OFFLINE After successfully activating SerialNET Auto-Link mode (!) or Deferred Connection mode.

Note: To terminate SerialNET mode, issue the ESC sequence (+++), power-cycle the W24 with the MSEL signal pulled low for less than 5 seconds, or pull the MSEL signal low for more than 5 seconds during runtime. After exiting SerialNET mode, W24 returns to normal AT+i command mode.



Web Server Interface

+iWWW - Activate Embedded Web Server

Syntax: AT+iWWW[:*n*]

Activates W24's internal web server.

Parameters: <*n*>=Web browser backlog. *n* represents the number of browsers that can connect to W24's internal web server simultaneously at any given time.

Note: Each additional browser (over the *n* number), will close a previous one and open its own.

Command Options: <*n*>=1..3

Default: <*n*>=1

Returns: **I**/(<Local IP addr>)
where,
<Local IP addr> is the W24 local IP address.

Note: If the web server is already open, then I/(<Local IP addr>) is returned without any action taken.

In a dial-up environment, W24 goes online and the <local IP addr> is assigned dynamically by the ISP.

In an WLAN environment, the IP address is assigned by a DHCP server or configured by the DIP parameter.

I/ERROR If connection to the Internet failed.

+iWNXT - Retrieve Next Changed Web Parameter

Syntax: AT+iWNXT

Retrieves the Parameter Tag name and new value of the next changed application web parameter, which has not been retrieved since it has been changed by the remote browser.

Returns: <Parameter Tag>=<New Value> <CR><LF>
When there are no more remaining changed parameters, a blank <CR><LF> terminated line is returned.

Followed by: <*n*>=1..3

I/O

File Transfer Protocol (FTP)

+i[@]FOPN - FTP Open Session

Syntax: AT+i[@]FOPN:<server>[,<port>]:<user>,<pass>[,<acct>]

Opens an FTP link to an FTP server.

Parameters:

<server> Logical name of the FTP or the server's IP address.

<port> Optional FTP port in the range 0..65535.

<user> FTP user's name.

<pass> FTP user's password.

<acct> Optional FTP account.

Command Options:

<server> The server name may be any legal Internet-server name, which can be resolved by the W24's DNS (Domain Name Server) settings. The server name may also be specified as an absolute IP address given in DOT form.

<port> Specifies the FTP server's listening port. If not specified, port 21 (decimal) is assumed.

<user> User's name string. This must be a registered user on the FTP server. Some servers allow anonymous login, in which case *user*=anonymous.

<pass> Password to authenticate user. If special characters are used, the password must be specified within quotes. It is customary that servers that allow anonymous login request an e-mail address as a password.

<acct> Some FTP servers require an account in order to allow a certain subset of the commands. In this case, the account name must be specified when opening the FTP link.

@ The optional @ is used to flag the Force PASV mode. When @ is specified, W24 only uses the PASV method when opening a data socket to server for FTP data transfer.

Result Code:

I/*<FTP handle>* Upon successfully connecting to the FTP Server and authenticating the user, a socket handle is returned. The handle *<FTP handle>* is used to reference the FTP session in all following FTP commands.

I/ERROR Otherwise.

+iFDL - FTP Directory Listing

Syntax: AT+iFDL:<F_hn>[,<path>]

Returns a full FTP directory listing.

Parameters:

<F_hn> An open FTP session handle.

<path> Directory or filename wild card.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Optional directory name or filename wild card. If <path> is a directory, that directory's files are listed. If it is a filename wild card, only matching filenames in the current directory are listed. If <path> is not specified, the current directory is listed in full.

Result Code:

I/OK To acknowledge successful receipt of the command.

I/ERROR If <F_hn> is not an open FTP session or otherwise some error has occurred.

Returns: A list of filenames with file attributes. Each file is listed on a separate line, terminated by <CR/LF>. The file data line syntax is FTP server-dependant.

Followed by:

I/ONLINE After successfully retrieving the directory list.

+iFDNL - FTP Directory Names Listing

Syntax: AT+iFDNL:<F_hn>[,<path>]

Returns the FTP directory name list.

Parameters:

<F_hn> An open FTP session handle.

<path> Optional directory or filename wild card.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Optional directory name or filename wild card. If <path> is a directory, that directory's files are listed. If it is a filename wild card, only matching filenames in the current directory are listed. If <path> is not specified, the current directory is listed in full.

Result Code:

I/OK	To acknowledge successful receipt of the command.
I/ERROR	If <i><F_hn></i> is not an open FTP session or otherwise some error has occurred.
Returns:	A bare list of filenames. Each file name is listed on a separate line, terminated by <CR/LF>. No attributes are returned in addition to the filename.
Followed by:	
I/ONLINE	After successfully retrieving the directory list.

+iFMKD - FTP Make Directory

Syntax: AT+iFMKD:<F_hn>,<path>

Creates a new directory on the FTP server's file system.

Parameters:

<F_hn> An open FTP session handle.

<path> Directory pathname.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Directory name. A new directory will be created under the current directory, as indicated by path. If path includes nonexistent subdirectories, some FTP servers will create them as well.

Result Code:

I/OK To acknowledge successful receipt of the command.

I/ERROR If *<F_hn>* is not an open FTP session or otherwise some error has occurred.

+iFCWD - FTP Change Working Directory

Syntax: AT+iFCWD:<F_hn>,<path>

Changes the current FTP working directory.

Parameters:

<F_hn> An open FTP session handle.

<path> New directory pathname.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Absolute or relative path name of the new directory. The special directory ".." signifies "one directory up".

Result Code:

I/OK After successfully changing the working directory.

I/ERROR Otherwise.

+iFSZ - FTP File Size

Syntax: AT+iFSZ:<F_hn>,<path>

Reports an FTP file size.

Parameters:

<F_hn> An open FTP session handle.

<path> File pathname.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Absolute or relative path name of the remote file.

Result Code:

I/<file size> W24 reports *path's* file size in bytes if the file exists and the FTP server supports the file size FTP command. Followed by: **I/OK**.

I/ERROR Otherwise.

+iFRCV - FTP Receive File

Syntax: AT+iFRCV:<F_hn>,<path>

Downloads a file from an FTP server.

Parameters:

<F_hn> An open FTP session handle.

<path> File pathname.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Absolute or relative path name of the remote file.

Result Code:

I/OK When command has been received and about to be processed.

I/ERROR If `<F_hn>` is not an open FTP session or otherwise some error has occurred.

Followed by:

I/ERROR If the FTP RECV command could not be processed.

-or- **I/**`<sz><CR><LF>`

Followed by: `<data stream>`

where,

`<sz>` is the exact size (in bytes) of the `<data stream>` to follow. If `<sz>` cannot be determined, W24 returns I/O followed by the data stream. When this is the case, the host must monitor for a timeout condition of at least 5 seconds without any data being transmitted before seeing the **I/ONLINE** to deduce that the data stream is complete.

If `<sz>` was reported but a transmission error occurred, preventing the W24 from returning all `<sz>` data bytes - an **I/ERROR** command is issued after a 5 seconds non-transmission period. See FTP Receive Flow Diagram.

Followed by:

I/ONLINE After successfully retrieving file contents.

+iFSTO - FTP Open File for Storage

Syntax: `AT+iFSTO:<F_hn>,<path>[,<sz>]`

Opens a remote FTP server file for upload.

Parameters:

`<F_hn>` An open FTP session handle.

`<path>` Destination file pathname.

`<sz>` Optional size in bytes to reserve for the file on the remote FTP server.

Command Options:

`<F_hn>` Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

`<path>` Absolute or relative path name of the remote destination file.

Following this command, data is transferred to the remote file using one or more +iFSND commands. The file transfer is complete by issuing a +iFCLF (FTP File Close) command.

Result Code:

I/OK If file `<path>` was successfully opened for writing on the FTP server.

I/ERROR Otherwise.

+iFAPN - FTP Open File for Appending

Syntax: AT+iFAPN:<F_hn>,<path>[,<sz>]

Opens an existing remote FTP server file for Append.

Parameters:

<F_hn> An open FTP session handle.
 <path> File pathname.
 <sz> Size in bytes to reserve for the file on the server.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.
 <path> Absolute or relative path name of the remote destination file.
 Following this command data is transferred to the remote file using one or more +iFSND commands. The file transfer is complete by issuing a +iFCLF (FTP File Close) command.

Result Code:

I/OK If file <path> was successfully opened for appending on the FTP server.
I/ERROR Otherwise.

+iFSND - FTP Send File Data

Syntax: AT+iFSND:<F_hn>,<sz>:<stream...>

Uploads data to a remote FTP server file. Valid only after a successful AT+iFSTO or AT+iFAPN command.

Parameters:

<F_hn> An open FTP session handle.
 <sz> The exact size of the data stream that follows.
 <stream> A byte stream of size <sz> composing the remote file contents.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<stream> An 8-bit byte stream of exactly size **<sz>**. If **<sz>** is larger than 256 bytes, W24 assumes host flow control. Depending on the setting of the FLW parameter, the flow control mode is either software or hardware. Under software flow control mode, the host processor must respond to W24's flow control characters. The flow control protocol is detailed in the "Host -> W24 Software Flow Control" section later in this document. When software flow control is active, it is recommended to set W24 to Echo-Off mode. Under hardware flow control, the ~CTS/~RTS RS232 control signals must be connected and the host must respond to W24's ~CTS signal. The host may send data only when the ~CTS signal is asserted (active low). Several consecutive +iFSND commands may be issued in sequence to create a larger aggregate of data to be sent. The file transfer is complete by issuing a +iFCLF (FTP Close File) command.

Result Code:

I/OK After **<sz>** bytes have been transferred successfully to the FTP data socket.
I/ERROR Otherwise.

+iFCLF - FTP Close File

Syntax: AT+iFCLF:<F_hn>

Closes a file downloaded to a remote FTP server. Only valid after a successful AT+iFSTO or AT+iFAPN command and optional AT+iFSND commands.

Parameters:

<F_hn> An open FTP session handle.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

Result Code:

I/OK After successfully closing the file.
I/ERROR Otherwise.

+iFDEL - FTP Delete File

Syntax: AT+iFDEL:<F_hn>,<path>

Deletes a remote FTP file.

Parameters:

<F_hn> An open FTP session handle.

<path> File pathname

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

<path> Absolute or relative pathname of the remote destination file to delete.

Result Code:

I/OK After successfully closing the file.

I/ERROR Otherwise.

+iFCLS - FTP Close Session

Syntax: AT+i[!]FCLS:<F_hn>

Closes the FTP link.

Parameters:

<F_hn> An open FTP session handle.

Command Options:

<F_hn> Must have been obtained by a previous execution of an AT+iFOPN command during the current Internet mode session.

! Stay online after completing the command.

Result Code:

I/OK When command has been received and about to be processed.

Followed by:

I/DONE When the FTP link was the last open socket and after successfully closing the FTP link. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully closing the FTP link, when additional sockets are still active or the stay online flag (!) is specified.

-or-

I/ERROR Otherwise.

Telnet Client

+iTOPN - Telnet Open Session

Syntax: AT+iTOPN:<server>

Opens a Telnet link (socket) to a Telnet server on port 23.

Parameters:

<server> Logical name of the Telnet server or the server's IP address.

Command Options:

<server> The server name can be any legal Internet Server name that can be resolved by W24's DNS (Domain Name Server) settings. The server name may also be specified as an absolute IP address given in DOT form.

Result Code:

I/OK Upon successfully connecting to the remote Telnet server.

I/ERROR Otherwise.

+iTRCV - Telnet Receive Data

Syntax: AT+iTRCV[:<max>]

Receives data from the Telnet server.

Parameters:

<max> Optionally specifies the maximum number of bytes to transfer.

Result Code:

I/ERROR If no Telnet session is open or otherwise some error has occurred.

Returns: **I/<sz>[:<binary data stream>]**
where,
<sz> is the exact size of the binary data stream to follow.
If the socket input buffer is empty, W24 returns I/O. In this case the (:) and <binary data stream> are omitted.
<sz> is guaranteed to be equal or less than <max>, when specified.

+iTSND - Telnet Send Data Line

Syntax: AT+iTSND:<data line>

Sends data to the remote Telnet server.

Parameters:

<data line> A line of data bytes to be sent to the Telnet server. W24 terminates the **<data line>** with a **<CR><LF>** and sends it to the Telnet server.

Command Options:

<data line> If the line to be sent incorporates W24 delimiter characters (, ; : = ; ~), **<data line>** must be enclosed in single (') or double (") quotes. AT+i command's terminating **<CR>** is considered a terminating quote, as well.

Result Code:

I/OK After the **<data line>** has been successfully sent to the Telnet server.

I/ERROR Otherwise.

+iTBSN[%] - Telnet Send a Byte Stream

Syntax: AT+iTBSN[%]:<sz>:<stream>

Sends a byte stream of size **<sz>** to the Telnet server.

Parameters:

<sz> The exact size of the byte stream that follows.

<stream> A byte stream of size **<sz>** to be sent to the Telnet server.

Command Options:

<sz> 0..4GB

<stream> An 8-bit byte stream of exactly size **<sz>**. If **<sz>** is larger than 256 bytes, W24 assumes host flow control. Depending on the setting of the FLW parameter, the flow control mode is either software or hardware. Under software flow control mode, the host processor must respond to W24's flow control characters. The flow control protocol is detailed in the "Host -> W24 Software Flow Control" section later in this document. Under hardware flow control, the ~CTS/~RTS RS232 control signals must be connected and the host must respond to W24's ~CTS signal. The host may send data only when the ~CTS signal is asserted (active low).

% When the auto-flush (%) flag is specified, the Telnet socket is automatically flushed immediately after receiving the **<stream>** from the host. Otherwise, data will be transmitted to the Internet only in integral quantities of the specified Maximum Transfer Unit (MTU) or when the AT+iTFSSH command is issued.

Result Code:

I/OK After **<sz>** bytes have been transferred successfully to the Telnet socket's output buffer.

I/ERROR Otherwise.

+iTFSH[%] - Flush Telnet Socket's Outbound Data

Syntax: AT+iTFSH[%]

Flushes (immediately sends) all the data accumulated in a Telnet socket's outbound buffer.

Command Options:

% When the flush-and-acknowledge ('%') flag is specified, W24 flushes and waits for the Telnet server receipt acknowledgment of all outstanding outbound data.

Result Code:

I/OK If all outbound data has been received and acknowledged by the Telnet server.

I/ERROR Otherwise.

+iTCLS - Telnet Close Session

Syntax: AT+i[!]TCLS

Closes the Telnet link.

Command Options:

! Stay online after completing the command.

Result Code:

I/OK If an active Telnet socket exists.

Followed by:

I/DONE When the Telnet link was the last open socket and after successfully closing the Telnet link. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully closing the Telnet link, when additional sockets are still active or the stay online flag (!) is specified.

-or-

I/ERROR Otherwise.

Direct Socket Interface

+iSTCP - Open and Connect a TCP Socket

Syntax: AT+iSTCP:<host>,<port>[,<lport>]

Opens a Transmission Control Protocol (TCP) client socket and attempts to connect it to the specified <port> on a server defined by <host>.

Parameters:

<host> Logical name of the target server or a host IP address.

<port> 0..65535, target port.

<lport> Optional local port on W24.

Command Options:

<host> The server name may be any legal Internet server name that can be resolved by W24's DNS (Domain Name Server) settings. The server name can also be specified as an absolute IP address given in DOT form.

<port> It is assumed that the server system is listening on the specified port.

<lport> Can be optionally specified to force W24 to use *lport* as the local port when opening the TCP socket. If unspecified, W24 allocates a port from its internal pool¹.

Result Code:

I/*<sock handle>* Upon successfully opening and connecting the TCP socket to the <host>:<port>, a socket handle is returned. The socket handle *<sock handle>* is in the range 0..9 and used to reference the socket in all following socket commands.

I/ERROR Otherwise.

The Socket Command Abort may be used to abort prematurely.

Note: ¹ W24 uses the port range [1025 .. 2048] when assigning default local ports. The host should refrain from specifying local ports in this range to ensure that Error 218 is not generated as a result of requesting local ports that overlap internal assignments.

+iSUDP - Open a Connectionless UDP Socket

Syntax: AT+iSUDP:<host>,<rport>[,<lport>]

Opens a UDP (User Datagram Protocol) socket and sets the remote system's <host>:<port> address.

Parameters:

<host> Logical name of the target server or a host IP address, or 0.0.0.0 to open a non-connected socket.

<rport> Remote port number to send to, or 0 to open a non-connected socket.

<lport> Optional local UDP port to use.

Command Options:

<host> The remote system's name may be any legal Internet server name that can be resolved by W24's DNS (Domain Name Server) settings. The server name may also be specified as an absolute IP address given in DOT form. When the **<host>** is defined, the resulting UDP socket is created and connected. If **<host>**=0.0.0.0, the socket is created but remains unconnected. The first UDP packet to arrive automatically latches the sender's IP port, in effect connecting the socket.

<rport> Specifies the remote system's port.

<lport> Specifies the local port to use. If unspecified, W24 allocates a port from its internal pool.

Result Code:

I/<sock handle> Upon successfully opening and connecting the UDP socket to **<host>:<port>**, a socket handle is returned. The socket handle **<sock handle>** is in the range 0..9 and used to reference the socket in all following socket commands.

I/ERROR Otherwise.

The Socket Command Abort may be used to abort prematurely.

+iLTCP - Open a TCP Listening Socket

Syntax: AT+iLTCP:<port>,<backlog>

Opens a TCP listening socket on the local IP address and the specified port **<port>**. The **<backlog>** parameter specifies the maximum number of remote concurrent connections allowed through the listening socket.

Parameters:

<port> 0..65535

<backlog> 1..10

Command Options:

<port> Listening port to be used by a remote system when connecting to W24.

<backlog> Specifies the maximum number of active connections that may be concurrently established through the listening socket. Once the listening socket is open, it automatically *accepts* remote *connect* requests up to the maximum allowed. When a remote system connects through the listening socket, a new TCP socket is spawned internally ready to send and receive data. See the AT+iLSST command for details on retrieving the handles of active sockets connected through a listening socket. When a connected socket is closed by the host using the AT+iSCLS command, the listening socket allows a new connection in its place..

Result Code:

I/*<sock handle>* Upon successfully opening a TCP listening socket, a socket handle is returned. The socket handle *<sock handle>* is in the range 10..11 and used to reference the socket in all following socket commands.

I/ERROR Otherwise.

+iLSST - Get a Listening Socket's Active Connection Status

Syntax: AT+iLSST:*<hn>*

Retrieves handles of active socket connections established through the listening socket identified by *<hn>*.

Parameters:

<hn> A TCP listening socket handle of an open listening socket.

Command Options:

<hn> Must have been obtained by a previous AT+iLTCIP command during the current Internet session.

Result Code:

I/(*<hn₁>*,...,*<hnBacklog>*) A list of active socket handles. The list contains *<backlog>* elements, where *<backlog>* was used when opening the listening socket identified by *<hn>*.
Where,
<hn_i> ≥ 0 : A handle to an active connected socket
 $= -1$: No connection has been established.

I/ERROR If *<hn>* is not an open listening socket, or otherwise some error occurred.

+iSST - Get a Single Socket Status Report

Syntax: AT+iSST:*<hn>*

Retrieves a socket status report for a single socket. This is a subset of the general AT+iRP4 report command.

Parameters:

<hn> A TCP/UDP socket handle.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket *accepted* by a listening socket.

Result Code:

II(<sockstat>) where,
sockstat >=0 - Number of bytes pending in socket <hn>'s input buffer.
sockstat <0 - Socket error code.

I/ERROR If some error occurred.

+iSCS - Get a Socket Connection Status Report

Syntax: AT+iSCS:<hn>

Retrieves a socket's connection status report without reporting the number of buffered characters.

Parameters:

<hn> A TCP/UDP socket handle.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket **accepted** by a listening socket.

Result Code:

II(<sockstat>) where,
sockstat=000 - Socket is connected without any associated errors.
sockstat<0 - Socket error code.

I/ERROR If some error occurred.

+iSSND[%] - Send a Byte Stream to a Socket

Syntax: AT+iSSND[%]:<hn>,<sz>:<stream>[<checksum>]

Sends a byte stream of size *sz* to the socket specified by the socket handle *hn*.

Parameters:

- <hn> A TCP/UDP socket handle of an open socket.
- <sz> The exact size of the byte stream that follows.
- <stream> A byte stream of size *sz* to be sent to the specified socket. When W24 is in checksum mode (CKSM set to 1) or when sending data over an SSL socket, *sz* is limited to 2048 bytes.
- <checksum> A two-byte checksum. Checksum is calculated by summing all the characters in *stream* modulo 65536 and taking two's complement of the result. Checksum is sent as big-endian. This parameter must be appended by the host application when W24 is in checksum mode.

Command Options:

- <hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket **accepted** by a listening socket.
- <sz> 0..4GB
- <stream> An 8-bit byte stream of exactly size *sz*. If *sz* is larger than 256 bytes, W24 assumes host flow control. Depending on the setting of the FLW parameter, the flow control mode is either software or hardware.
Under software flow control mode, the host processor must respond to W24's flow control characters. The flow control protocol is detailed in the "Host -> W24 Software Flow Control" section. Under hardware flow control, the ~CTS/~RTS RS232 control signals must be connected and the host must respond to W24's ~CTS signal. The host may send data only when the ~CTS signal is asserted (active low).
- % When the auto flush (%) flag is specified, the socket is automatically flushed immediately after receiving the stream. Otherwise, data is transmitted to the Internet only in integral quantities of the specified Maximum Transfer Unit (MTU) or when the AT+iSFSH command is issued.

Result Code:

- I/OK<CR><LF><CR><LF>** After *sz* bytes have been transferred successfully to the socket's output buffer.
- I/ERROR** Otherwise.

Note: When W24 is in checksum mode, it calculates the checksum of the data received from host and compares it with checksum sent by host. If the two match, the result code is **I/OK**. Otherwise, **I/ERROR (228)** is returned and the data discarded. If host attempts to send more than 2048 bytes, **I/ERROR (227)** is returned.

The Socket Command Abort may be used to abort prematurely.

+iSRCV - Receive a Byte Stream from a Socket's Input Buffer

Syntax: AT+iSRCV:<hn>[,<max>]

Receives a byte stream from the TCP/UDP socket specified by the socket handle *hn*. Received data is valid only if it already resides in W24's socket input buffer at the time this command is issued.

Parameters:

- <hn> A TCP/UDP socket handle of an open socket.
- <max> Optionally specifies the maximum number of bytes to transfer. Additional bytes may remain in the socket input buffer following this command.

Command Options:

- <hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket *accepted* by a listening socket.
- <max> If <max> is not specified, all available bytes residing in the socket input buffer are returned.

Returns:

I<sz>[:<stream>][<checksum>]

where,
sz is the exact size of the binary data stream to follow.
 If the socket input buffer is empty, W24 returns **I/O<CR><LF>**. In this case, *stream* is omitted.
sz is guaranteed to be equal or less-than *max*, when specified.
checksum is a two-byte checksum. This parameter is calculated by W24 only when it is in checksum mode (CKSM set to '1').
 checksum is calculated by summing all the characters in stream modulo 65536 and taking two's complement of the result.
 checksum is sent as big-endian. The host application is assumed to calculate its own checksum upon receipt of stream and compare it against the checksum bytes received from W24. If the two checksums don't match, the host can issue an AT+!SRCV command, which causes W24 to re-transmit the data. The next AT+iSRCV command that the host issues causes W24 to dump all data transmitted to host in the previous AT+iSRCV command.

I/ERROR If <hn> is not an open socket, or otherwise some error occurred.

+iGPNM - Get Peer Name for a Specified Socket

Syntax: AT+iGPNM:<hn>

Retrieves peer name (<IP>:<Port>) of a remote connection to a TCP/UDP socket specified by the socket handle <hn>.

Parameters:

<hn> A TCP/UDP socket handle of an open socket.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket *accepted* by a listening socket.

Result Code:

I(<IP>:<Port>) where,
<IP> is the remote peer's IP address, and **<Port>** is the remote peer's port for this connection.

I/ERROR If **<hn>** is not an open socket handle, or otherwise some error occurred.

+iSDMP - Dump Socket Buffer

Syntax: AT+iSDMP:<hn>

Dumps all buffered data currently accumulated in a socket's inbound buffer. The socket remains open.

Parameters:

<hn> A TCP/UDP socket handle of an open socket.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket *accepted* by a listening socket.

Result Code:

I/OK If **<hn>** is a handle to an open socket.

I/ERROR Otherwise.

+iSFSH[%] - Flush Socket's Outbound Data

Syntax: AT+iSFSH[%]:<hn>

Flushes (immediately sends) accumulated data in a socket's outbound buffer.

Parameters:

<hn> A TCP/UDP socket handle of an open socket.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket *accepted* by a listening socket.

% When the flush-and-acknowledge (%) flag is specified and *<hn>* is a TCP socket handle, W24 flushes and waits for the peer receipt acknowledgment of all outstanding outbound data. Common errors associated with this flag are 215 (carrier lost) and 203 (socket closed by peer in an orderly manner or did not receive ACK after repeated attempts to retransmit unacknowledged data).

Result Code:

I/OK If *<hn>* is a handle to an open socket and, when *<hn>* is a TCP socket handle, all outbound data has been received (and when (%) flag specified also acknowledged) by peer.

I/ERROR Otherwise.

The Socket Command Abort may be used to abort prematurely.

+iSCLS - Close Socket

Syntax: AT+i[!]SCLS:*<hn>*

Closes a TCP/UDP socket.

If the socket is the only open socket and the stay online flag (!) is not specified, W24 terminates the Internet session and goes offline.

Parameters:

<hn> A TCP/UDP socket handle of an open socket.

Command Options:

<hn> Must have been obtained by a previous execution of an AT+iLTCP, AT+iSTCP or AT+iSUDP command during the current Internet mode session. Or a socket accepted by a listening socket.

A socket is always flushed before being closed. TCP sockets are disconnected from the remote host server in an orderly manner.

! Stay online after completing the command.

Result Code:

I/OK If *<hn>* is a handle to an open socket.

I/ERROR Otherwise.

Followed by:

I/DONE After successfully closing the last open socket. Allow a 2.5 seconds delay for W24 re-initialization following an Internet mode session.

-or-

I/ONLINE After successfully closing the socket, while additional sockets are still open or if the stay online flag (!) is specified.

-or-

I/ERROR Otherwise.

Special Modem Commands

+iMCM - Issue Intermediate Command to Modem

Syntax: AT+iMCM[:<AT command>]

Sends a single AT command to the modem during an internet session or enters Modem Command mode.

Parameters:

<AT command> Optional single AT command to be sent to modem.

Command Options:

<AT command> W24 puts the modem in command mode by issuing the (+++) escape sequence and then sends <AT command> to the modem, followed by a <CR>. <AT command> must include the AT prefix. After receiving the modem's response, W24 restores the modem to online operation mode by issuing the ATO command. If <AT command> is not specified, W24 enters Modem Command mode. In this mode, all following commands are transferred as-is to the modem. Modem replies are relayed back to the host processor. W24 does not translate the commands. Modem Command mode is exited after the host issues the ATO command. W24 transfers the ATO command to the modem and relays the modem's response back to the host.

Returns: Modem's responses including command echo, if enabled.

Followed by:

I/OK When the modem successfully returns online.

I/ERROR If modem was unable to go back online.

Wireless LAN Mode

The W24 includes a Wireless LAN driver for the Marvell 88W8686 802.11b/g WiFi chipset. In addition, the W24 firmware contains WEP and WPA encryption of WPA-PSK with TKIP and WPA2-PSK with AES for this chipset.

WPA security requires a parameter that contains the Personal Shared Key (PSK), sometimes referred to as the passphrase. The Wireless LAN Passphrase (WLPP) parameter is used to set the passphrase. When passphrase contains a value, W24 uses WPA security when connecting to an Access Point (AP). Note, however, that for WPA-PSK to be active, an *SSID* (+iWLSI parameter) must also be defined. This parameter has precedence over WEP parameters. In other words, when WLPP contains a value (and WLSI is defined) WPA is used - even if WEP parameters are defined. The maximum allowable wireless LAN transmission rate is determined by the WLTR command.

The type of WPA protocol to be used is determined by the value of the WSEC parameter: a '0' value means the WPA-TKIP protocol will be used, whereas a '1' value specifies the WPA2-AES protocol.

Several commands, listed below, enable W24 to control the operation of the Marvell WiFi chipset.

+iWLTR - Wireless LAN Transmission Rate

Syntax: AT+iWLTR=<tr>

Sets the maximum allowable wireless LAN transmission rate. After a SW reset, WLTR returns to its default value (54 Mbps).

Parameters:

<tr> tr=0..13

Command Options:

tr=0	Maximum supported transmission rate (54 MBps)
tr=1	Limited to 1 Mbps
tr=2	Limited to 2 Mbps
tr=3	Limited to 5.5 Mbps
tr=4	Limited to 11 Mbps
tr=5	<i>Reserved</i>
tr=6	Limited to 6 Mbps
tr=7	Limited to 9 Mbps
tr=8	Limited to 12 Mbps
tr=9	Limited to 18 Mbps
tr=10	Limited to 24 Mbps
tr=11	Limited to 36 Mbps

tr=12 Limited to 48 Mbps

tr=13 Limited to 54 Mbps

Default: 0 (Maximum transmission rate)

Result Code:

I/OK If *tr*=0..13.

I/ERROR Otherwise.

+iWLPW - Set WLAN Tx Power

Syntax: AT+iWLPW=<*n*>

Sets the transmission power of the Marvell WLAN chipset.

Parameters: *n*=0-20

n=0 Use Marvell's automatic power level adaptation scheme.

n=1-20 Set a fixed transmission power level.

Default: *n*=0

Result Code:

I/OK If power set succeeded.

I/ERROR (042) If *n* is an illegal value.

-or-

I/ERROR (402) If power set failed.

+iWRFU - WLAN Radio Up

Syntax: AT+iWRFU

Turns on radio transmission of the Marvell WLAN chipset.

Parameters: None

Result Code:

I/OK If operation succeeded.

I/ERROR (403) Otherwise.

+iWRST - Reset WLAN Chipset

Syntax:	AT+iWRST
	Performs a hardware reset of the Marvell WLAN chipset.
Parameters:	None
Result Code:	
I/OK	If operation succeeded.
I/ERROR (404)	Otherwise.

+iWLBM - WLAN B Mode

Syntax:	AT+iWLBM
	Sets the Marvell WLAN chipset to 802.11/b mode. Allowable Tx transmission rates for this mode are: 1, 2, 5.5 and 11 Mbps.
Parameters:	None
Result Code:	
I/OK	Always.

+iWLGM - WLAN G Mode

Syntax:	AT+iWLGM
	Sets the Marvell WLAN chipset to 802.11/g mode. Allowable Tx transmission rates for this mode are: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps.
Parameters:	None
Result Code:	
I/OK	Always.

Roaming Mode

When set to operate in Roaming mode, W24 can roam seamlessly among Access Points (APs) sharing the same SSID and the same security configuration without interrupting its IP connectivity. W24 also has a monitoring mechanism that is sensitive to drops in AP signal strength. When W24 detects such a drop, it automatically starts searching for APs in its vicinity that have a stronger signal, while remaining connected to the current AP.

The following parameters are required to set W24 to Roaming mode:

- WROM - Enables Roaming mode.
- WPSI - Sets the time interval between consecutive scans that W24 performs for APs in its vicinity.
- WSRL - Sets a low SNR threshold for W24 in Roaming mode.
- WSRH - Sets a high SNR threshold for W24 in Roaming mode.

In addition, two reports provide useful information pertaining to the Roaming feature:

- AT+i!RP10 - Returns a report of the current WLAN connection.
- AT+iRP20 - Returns a list of all APs and ad-hoc networks available in the vicinity.

W24 Behavior Following a Hardware or Software Reset

After power-up, hardware or software reset, W24 starts scanning for APs in its vicinity at intervals set by the WPSI parameter. W24 reads the value set in the WLSI parameter and acts accordingly:

- If WLSI refers to an AP, W24 scans for all APs in its vicinity. W24 attempts to connect to an AP whose SSID is listed first in the WSIn parameter. If several APs having that same SSID exist, W24 attempts to connect to the one having the strongest signal. If association succeeds, W24 stops scanning and activates its DHCP client. It then monitors the SNR level of the AP it is associated with.
- If WLSI refers to an ad-hoc network, W24 scans for all ad-hoc networks in its vicinity. W24 attempts to join an ad-hoc network whose SSID is listed first in the WSIn parameter. If no such network is found, W24 creates its own network and stops scanning.
- If WLSI is set to (*), W24 stops scanning and remains disconnected.

W24 Behavior when AP Signal Becomes Weak

When the beacon signal of the AP with which W24 is associated becomes weak (SNR drops below the level set by the WSRL parameter), W24 starts its periodic scan for APs having SNR above the threshold set by the WSRH parameter.

W24 attempts to connect to the AP that appears first on the list of SSIDs specified in the WSIn parameter, while remaining connected to the current AP. If association with the new AP fails, W24 continues scanning until it succeeds connecting to an AP with a stronger signal.

When in Roaming mode, W24 does not restart its DHCP client process for new connections.

When W24 is *not* in Roaming mode, W24 remains connected to an AP as long as it has an open active socket, or until triggered by a Link Lost event. When not in Roaming mode, W24 ignores any decrease in AP signal strength while having open active sockets.

When W24 is *not* in Roaming mode and no active sockets are open, W24 starts periodic scanning for APs having an SNR level above the WSRH threshold. W24 attempts to connect to the AP that has the highest priority. After associating with an AP, W24 starts its DHCP client and monitors the SNR level of the AP it is associated with.

W24 Behavior in the Event of a Lost Link

If the connection is *not* active, W24 starts periodic scanning for APs and attempts to connect to an AP having the highest priority. After associating to an AP, W24 starts its DHCP client and monitors the SNR level of the AP it is associated with.

If the connection is active, W24 waits for an IP activity command from the host. When such a command is sent, W24 performs a software reset and starts scanning for APs. W24 responds with ERROR (074) to indicate that the current connection has been lost.

Multiple SSIDs

The Multiple SSIDs feature allows you to define an ordered list of SSIDs of Access Points (APs) or ad-hoc networks with which W24 attempts to connect upon power-up. Each SSID listed can have one of the following security types:

- WEP-64
- WEP-128
- WPA-TKIP
- WPA2-AES
- No security

The following parameters allow you to define multiple SSIDs:

- **WSIn** - Defines an ordered list of allowable SSIDs.
- **WPPn** - Sets the Wireless LAN PSK passphrase for WPA and WPA2 encryption for each individual SSID on the list.
- **WKYn** - Sets the Wireless LAN WEP key for each individual SSID on the list.
- **WSTn** - Sets the Wireless LAN security type for each individual SSID on the list.

W24 Power Save Mode

W24 has a Power Save mode for achieving energy savings. You enable Power Save mode by setting the PSE parameter to any value *n* between 1 and 255 seconds. When *n* seconds have elapsed without any activity on the host or modem serial ports, W24 shuts down most of its circuits. Renewed activity on the serial ports, or incoming data from the WLAN, restores W24 to full operational mode.

If, in addition, the WLPS parameter is set to any value *m* between 1 and 5, W24 can force the Marvell WiFi chipset into either Power Save or Deep Sleep mode:

- If W24 is currently associated with an AP, or is configured to operate in ad-hoc mode, W24 will force the Marvell chipset into Power Save mode. In Power Save mode, the Marvell chipset will go to sleep for *m* beacon periods when no communication has taken place (command, Tx, or Rx activity) for one full beacon period.
- If W24 is not associated with an AP, W24 will force the Marvell chipset into Deep Sleep mode. W24 will perform a periodic scan every *p* seconds, as set by the WPSI parameter, for APs in its vicinity. If it fails to locate and associate with an AP, it will wait for *n* seconds, as set by the PSE parameter, before forcing the Marvell chipset back to Deep Sleep.

IP Registration

When W24 goes online in a dial-up environment, it is normally assigned a dynamic IP address during PPP establishment. Since a different IP address is usually assigned every session, it is not practical to use W24 as a server, since the clients do not know what IP address to use.

Furthermore, under these restrictions, there is no practical way to know whether a specific system is online or offline. A similar problem occurs when using the W24 WLAN, which is configured to use a DHCP server. In this environment, a different IP address is usually assigned every time the W24 boots and connects to the WLAN.

To overcome this problem, W24 incorporates built-in procedures designed to register its IP address on a server system each time it goes online. Once registered, client systems may interrogate the servers in order to verify the online status of a specific system and retrieve its currently assigned IP address. The IP registration process is governed by several AT+i parameters. Once these parameters are configured, W24 registers its IP address accordingly when it goes online as a result of an explicit AT+i command (AT+iUP) or as a result of automated Internet session establishment procedures, such as a triggered Internet session or when going online as a SerialNET mode server.

In cases where W24 uses a NAT gateway to the Internet, it can be configured to register the NAT's IP address and a special port that is linked to W24 in the NAT's configuration. See details in the RRRL parameter description. When this is the case, the RRRL parameters (IP and port) are used instead of the local IP and port values that W24 is assigned, in all registration methods (RRMA, RRSV, and RRWS).

W24 includes several IP registration methods, as described below.

E-Mail Registration

W24 registers itself by sending an e-mail that contains its ID information and current IP address. When the RRMA parameter contains an e-mail address, W24 sends an e-mail containing its current IP address or its RRRL to the address defined in RRMA during the registration procedure. The syntax of the e-mail body is:

<BDY parameter contents>

```
iChip-<D/L/S> S/N:<RP5> Version:<RP1> HN:<HSTN> IP:<IPA or RRRL>
Port:<LPRT or 80 or 0> http:// <IPA or RRRL><CR><LF>
```

The subject line of the e-mail is:

```
"RING RESPONSE LINK From: iChip-<D/L/S> S/N:<RP5> Version:<F/W
ver> HN:<HSTN> IP:<IPA or RRRL> Port:<LPRT or 80 or 0>"
```

where,

Port is LPRT if in SerialNET mode; 80 if not in SerialNET mode and AWS is enabled, and 0 if not in SerialNET mode and AWS=0. The receiving end may refer to the contents of the subject line to filter out this e-mail message.

Socket Registration

W24 registers itself by opening a socket to a registration server and sending its ID information and current IP address. When W24's RRSV parameter contains a value, W24 establishes a socket

to the server defined in RRSV during the registration procedure. When a socket is established, W24 transmits its ID information and current IP address (or the RRRL) in the following format:

```
"iChip-<D/L/S> S/N:<RP5> version: <RP1> HN:<HSTN> IP:<IPA or  
RRRL> Port:<LPRT or 80 or 0>"
```

The registration socket is then closed.

Web Server Registration

W24 registers itself by surfing to a web server with its ID information and current IP address as parameters.

If the RRWS parameter contains a URL (of a registration web server), W24 registers its ID information and IP using the URL by issuing a GET command along with a fixed format parameter line:

```
"<RRWS path>?SN=<RP5>&IP=<IPA or RRRL>&Wpt=<0 or the port defined  
in RRRL>&HN=<HSTN>" .
```

The web server must contain a CGI, .asp page, exe, etc., which make use of these parameters to register the W24.

If several registration parameters are configured, W24 goes through multiple registration processes. If more than one registration process fails, W24 returns an I/ERROR describing the first failure encountered. If all registrations fail, W24 returns I/ERROR(90).

DHCP Client

A DHCP client component in W24 in WLAN mode supports IP and server name acquisition from a standard DHCP Server. The W24 device attempts to contact and acquire server names from a DHCP server if and when its DIP (Default IP) parameter contains the special value 0.0.0.0.

When the DHCP acquisition procedure is successful, the W24's IPA (IP Address) parameter contains the assigned IP address retrieved from the DHCP server. In addition, server names relevant to W24 parameters are retrieved from the DHCP server, if and only if they contain empty values at power-up (see [Table 2-2](#)). Parameters that contain non-empty values retain those values. In addition, DNS values retrieved from the DHCP server are retained as additional alternative DNS addresses when DNS n contain user-defined values.

Table 2-2: Server Names Acquired from DHCP Server

Parameter Name	Function	Empty Value
IPG	Gateway	0.0.0.0
SNET	Subnet Mask	0.0.0.0
DNS1	Primary Domain Name Server	0.0.0.0
DNS2	Secondary Domain Name Server	0.0.0.0
SMTP	Email Send Server	" (Empty String)
POP3	Email Receive Server	" (Empty String)

All values acquired from the DHCP server are not retained as nonvolatile values. New values shall be acquired during the next DHCP session, which will be activated during the next W24 power-up, following a soft or hard reset or after the DHCP lease expires.

The DHCP client has two associated points in time when the DHCP server is contacted for additional negotiations. At T1 (usually after half the original lease period), W24 attempts to renew the lease period. If the renewal procedure fails, at T2 (usually after 7/8 the original lease period) W24 attempts to re-negotiate the lease. If the procedures at T1 and T2 fail and the lease expires, W24 continuously tries to locate a DHCP server for re-negotiation. When this is the case, W24 stores 0.0.0.0 in the IPA parameter and cannot communicate on the WLAN until a DHCP server is found and IP and server names are acquired.

DHCP Server

W24's DHCP server allows it to manage a network segment when no DHCP server is available. When W24 is configured to operate in iRouter mode, it provides access to the public internet via its modem connection. The DHCP server can handle up to 255 IP addresses concurrently.

Two parameters govern DHCP server functionality:

- **DPSZ:** The DHCP pool size parameter determines the range of IP addresses that W24 allocates for its clients.
- **DSLT:** The DHCP server lease time determines the lease time that W24 grants when assigning IP addresses.

The DHCP server is activated under the following conditions:

- An IP address is defined by the DIP parameter.
- The DPSZ parameter is set to a value greater than 0.
- Following a software reset (AT+iDOWN).

When activated, W24's DHCP server assigns IP addresses starting from DIP+1 up to DIP+DPSZ. In addition, the DHCP server offers the IP address stored in the IPG parameter as a gateway to clients, and the mask address stored in its SNET parameter as a Sub-Net. The assignment policy of W24 is as follows:

1. W24 attempts to assign the same IP for the same MAC address.
2. W24 starts re-using addresses only after using all the addresses in the pool.
3. W24 attempts to re-use the oldest expired address first.
4. W24 attempts to ping the address it is about to assign in order to avoid assigning an address already used.
5. W24 offers its SNET parameter as a Sub-Net. If SNET is 0.0.0.0, W24 calculates a new one according to address class.
6. W24 offers its IPG parameter as a gateway. If IPG is 0.0.0.0, W24 offers its IP address as a gateway.
7. W24 offers the primary IP address of the Domain Name Server stored in its DNS1 parameter to the client, provided it is not 0.0.0.0.

iRouter Mode

Introduction

W24's iRouter mode is used to provide a gateway to a multitude of WiFi devices through a single dialup or cellular link. In this configuration, W24's DHCP server may be used to assign IP addresses to the local hosts on the WiFi side. W24 also uses a Network Address Translator (NAT) to translate between local and public IP addresses.

While routing IP packets, W24 also accepts AT+i commands, as during normal operation. The CPF (Communication Platform) parameter selects which interface to use for Internet-related AT+i commands.

The following parameters and commands are used to configure iRouter mode behavior:

- Automatic Router Start (ARS) parameter - When set to 1, this parameter causes W24 to go online in iRouter mode upon power-up and start routing packets.
- Inactivity Timeout (IATO) parameter - When in iRouter mode, if no routing activity is detected for the period of time specified by this parameter, W24 disconnects its modem/cellular side and goes offline. After going offline and if ARS=1, W24 will go online and continue routing when the next packet that requires routing arrives.
- Start Router (STRR) command - Causes W24 to enter iRouter mode, go online on the dialup/cellular side, and start routing packets.
- Stop Router (STPR) command - Causes W24 to exit iRouter mode, go offline on the dialup/cellular side, and stop routing packets.

Establishing iRouter Mode

W24 can be entered into iRouter mode using one of two possible methods:

- When the ARS parameter is set to 1, automatically and immediately after power-up and after every soft reset induced by AT+iDOWN.
- By issuing the AT+iSTRR (Start Routing) command.

Upon entering iRouter mode, W24 immediately goes online on the dialup/cellular side. Packets are not buffered during dialup/cellular connection establishment. After establishing the connection, W24 starts the routing service.

Basic Routing

When W24 is in iRouter mode, it routes packets between its two communication platforms utilizing a Network Address Translator (NAT) to translate between the internal IP address space used on the WiFi side and the real IP address used on the dialup/cellular side.

The NAT translates internal IP addresses of outgoing packets to the real IP address space and makes the reciprocal translation of packets received in response.

Note: When using an FTP client to connect to an external FTP server through the iRouter, you must use the FTP client in passive mode. For example, if the FTP client is a W24, you must open the FTP session using AT+i@FTP.

Configuring W24 when in iRouter Mode

iRouter mode is terminated by any of the following occurrences:

- By issuing the AT+iSTPR (Stop Routing) command. When W24 receives this command, routing services are stopped and W24 goes offline on the dialup/cellular side. If ARS=1 (Auto Routing), W24 automatically goes online and restores routing services when the next packet arrives.
- Automatically after an idle time period (with no routing activity) has passed. The idle time period is defined in the IATO (Inactivity Timeout) parameter. Idle time terminates routing only if IATO has a positive value larger than 0. When IATO=0, idle time termination is effectively disabled. If ARS=1 (Auto Routing), W24 automatically goes online and restores routing services when the next packet arrives.
- By issuing the (+++) ESC string. W24 terminates iRouter mode and goes offline on the dialup/cellular side. Following an ESC sequence termination, W24 does not restore routing services even if ARS=1. To restore routing, either issue the AT+iSTRR command or, alternatively, if ARS=1- issue AT+iDOWN.

Configuring W24 when in iRouter Mode

While in iRouter mode, W24 can be configured using the same methods for W24 in general (see [Figure 2-2](#)):

- Assuming W24's website is enabled on the WiFi end, W24's internal configuration website can be accessed by any browser that is connected to the same WiFi network.
- Assuming W24's website is enabled on the dialup/cellular side, W24's internal configuration website can be accessed by any remote browser connecting to W24's port 80 over its public IP address.
- AT+i commands coming from the host application.

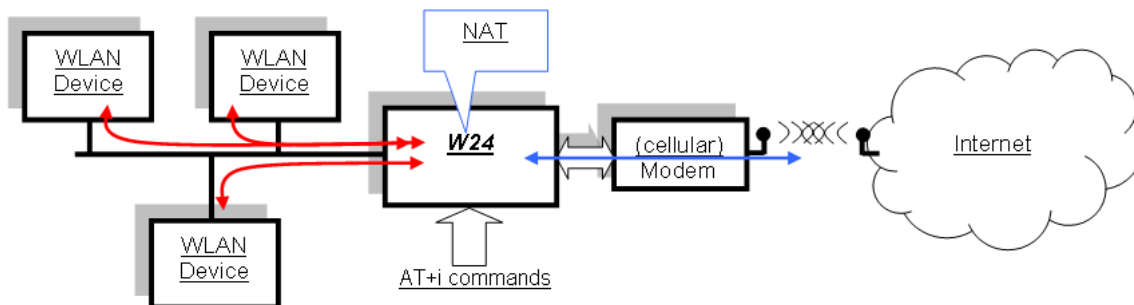


Figure 2-2: Configuring W24 when in iRouter Mode

AT+i Interface to W24

In addition to configuring the W24, AT+i commands can also be used to perform operations on either the WiFi or dialup/cellular communication platform (see [Figure 2-3](#)).

Using the CPF (Communication Platform) parameter, you can select either one of the communication platforms. When CPF=0, AT+i commands are directed towards the

dialup/cellular side; when CPF=1, they are directed towards the WiFi side. While processing AT+i commands, W24 continues to route packets seamlessly between the two platforms.

W24's responses to AT+i commands depend on the CPF value, as well. For example, the IP returned by AT+iIPA? command while CPF=1 is the WLAN-side IP.

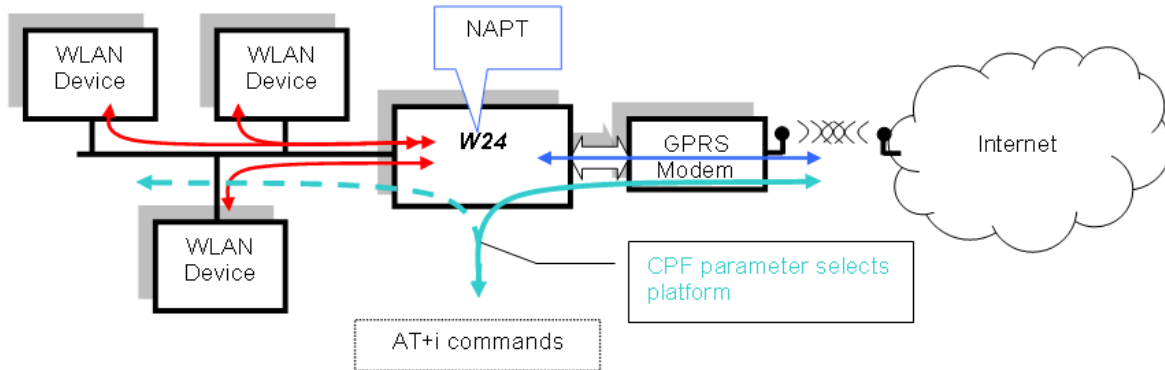


Figure 2-3: AT+i Interface to W24

Baud Rate Settings and Auto Baud Rate

iRouter mode supports all host and modem baud rates supported by W24. However, when auto routing is set (ARS=1), W24 does not support Auto Baud Rate. This is due to the fact that in iRouter mode, W24 starts routing packets immediately after power-up, and skips auto baud rate determination.

Therefore, when configuring W24 for auto routing (ARS=1), you must set a fixed baud rate in the BDRF parameter.

iRouter and Power Save Mode

W24 can be configured for Power Save mode while acting as a router. Note, however, that there is no buffering of packets in iRouter mode. The first packet arriving to W24 while in Power Save mode triggers W24 to wake up and go online on the cellular or dialup modem. Only after establishing a connection, does W24 start routing packets. The packets received during connection establishment are lost.

+iSTRR - Start Router

Syntax: AT+iSTRR

Causes W24 to immediately enter iRouter mode. Upon entering iRouter mode, W24 immediately goes online on the dialup/cellular side. Packets are not buffered during dialup/cellular connection establishment. After establishing the connection, W24 starts the routing service.

Result Code:

I/OK When command is received and about to be processed.

Followed by:

I/ONLINE After successfully going online on the dialup/cellular side.
I/ERROR Otherwise.

+iSTPR - Stop Router

Syntax: AT+iSTPR

Causes W24 to exit iRouter mode, go offline on the dialup/cellular side, and stop routing packets.

If ARS=1 (Auto Routing), W24 automatically goes online and restores routing services when the next packet arrives.

Result Code:

I/OK When command is received and about to be processed.

Followed by:

I/ONLINE After terminating the connection on the dialup/cellular side when CPF=1.

-or-

I/DONE After terminating the connection on the dialup/cellular side when CPF=0.

-or-

I/ERROR Otherwise.

Ad-Hoc Networks

An ad-hoc network is a Wireless Local Area Network (WLAN) in which some of the stations are part of the network only for the duration of a communications session or, in the case of mobile or portable devices, while in some close proximity to the rest of the network.

Ad-hoc networks do not require an Access Point (AP) to enable communication among stations. Each station can create a new ad-hoc network or join an existing one. Networks can freely merge into a single network or split into smaller ones, thus adapting to changing conditions such as topology, signal strength, and proximity to nearby ad-hoc networks. Combined with a W24 configured as an iRouter, an ad-hoc network can connect to the Internet through a dial-up or GPRS modem.

Configuration

Configuring the W24 to operate as a station in an ad-hoc network requires setting the following parameters:

- WLSI must be set to either '!' or '!'<SSID>'. When it is set to '!', W24 continuously searches for existing ad-hoc networks in its vicinity and joins the one having the strongest signal. When it is set to '!'<SSID>', W24 searches for an ad-hoc network having the specified Service Set Identifier (SSID). If it finds one it joins it, otherwise it creates a new network with this SSID.
- WLCH must be set to a default value. This value indicates the communication channel (1-13) to be used for beacon transmission in the ad-hoc network. When W24 joins an already existing network, it adopts the channel used by that network. If WLSI=!'<SSID>' and WLCH=0, W24 will only join an already existing network.

W24 Behavior in Ad-Hoc Mode

Automatic Scanning for Existing Ad-Hoc Networks

After power-up, W24 automatically attempts to locate and connect to an ad-hoc network, unless the WLSI parameter (SSID) is set to (*).

If the WLSI parameter contains an SSID string preceded by (!) or set to (!), W24 scans for ad-hoc networks only.

Creating a New Ad-Hoc Network

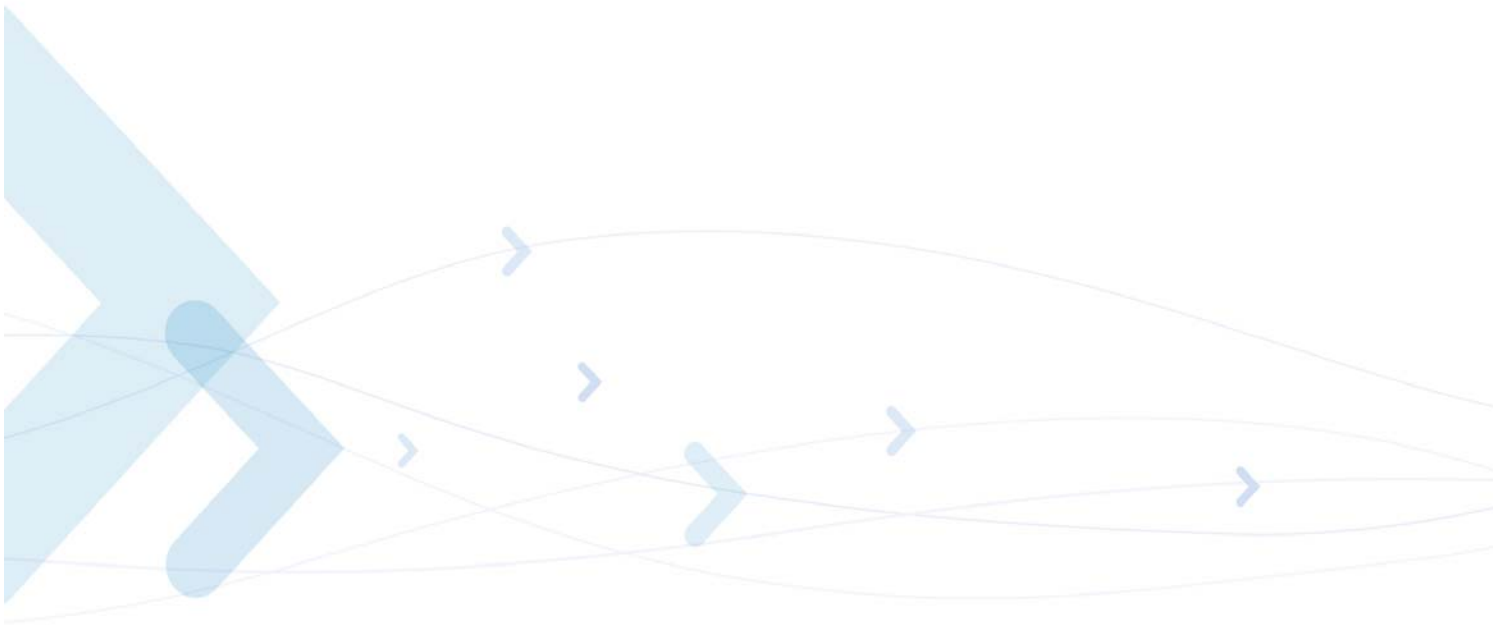
If W24 does not detect any ad-hoc networks in its vicinity, and the WLSI parameter contains an SSID, W24 creates a new ad-hoc network with its own BSSID.

Joining an Existing Ad-Hoc Network

If W24 detects ad-hoc networks in its vicinity and the WLSI parameter is set to (!), W24 joins the network having the strongest signal. Otherwise, W24 joins the network whose SSID is set by the WLSI parameter.

Merging Ad-Hoc Networks

When W24 is configured to operate in ad-hoc mode, the Marvell WiFi chipset it is connected to performs a periodic scan for other ad-hoc networks in the vicinity having the same SSID but a different BSSID. If a scan indicates the existence of such an ad-hoc network, the Marvell chipset initiates a procedure of merging networks. Networks merge into the one that was created earlier, provided they operate on the same channel.



Secure Socket Protocol

W24 supports the SSL3/TLS1 secure socket protocol, based on RFC2246. W24 supports the following Cipher suites:

- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

Establishing An SSL3/TLS1 Socket Connection

W24 supports a single SSL3/TLS1 TCP/IP active socket connection. Opening a secure socket on W24 involves two steps:

1. Open a standard TCP/IP socket to a secure server.
2. Initiate an SSL3/TLS1 handshake over the open socket to establish a secure session. SSL3/TLS1 handshake negotiations are initiated using the AT+iSSL command.

W24 negotiates the secure connection based on several security-related parameters. It authenticates the remote secure server by verifying that the server's certificate is signed by a trusted Certificate Authority (CA). The trusted CA's certificate is stored in W24's CA parameter. Following a successful SSL3/TLS1 handshake, W24 encrypts all data sent across the socket according to the cipher suite and keys agreed upon during the handshake. Data received on the socket is decrypted by W24 prior to making it available to the host processor.

Sending and Receiving Data over An SSL3/TLS1 Socket

The AT+iSSND command is used to send data over an SSL3/TLS1 socket, using the same syntax as for non-secure sockets:

```
AT+iSSND[%]:<hn>,<size>:<data>
```

However, the *size* parameter is interpreted as the size of the data packet to encrypt. It is limited to 2K. Receiving data on an SSL3/TLS1 socket is carried out using the AT+iSRCV command. W24 automatically decrypts data that arrives on the secure socket. The data transferred to the host is always decrypted data.

SSL3/TLS1 Handshake and Session Example

Take for example an SSL3/TLS1 server at `secure.sslserver.com` running a secure application on port 1503. Using W24, the following sequence opens a secure SSL3/TLS1 socket to that application and exchanges data securely. For clarity, commands sent to W24 appear in bold and W24 replies appear in *italics*.

AT+iSTCP:secure.sslserver.com,1503

I/000

Open a TCP/IP socket to a secure application.

W24 opens socket and returns handle 0.

AT+iSSL:0	W24 is instructed to negotiate an SSL3/TLS1 connection on socket handle 0.
<i>I/OK</i>	SSL3/TLS1 handshake was successful. SSL3/TLS1 connection established on socket handle 0.
AT+iSSND%:0,323:<...323 bytes of plain text data>	Host sends 323 bytes of plain text data via SSL3/TLS1 socket. W24 encrypts data and sends cipher text over the Internet. The '%' attribute indicates immediate flush.
<i>I/OK</i>	W24 encrypted and sent data.
AT+iRP4	Request socket status.
<i>I/(1267,-200,-200,-200,-200,-200,-200,-200,-200,-200)</i>	Socket 0 has 1267 plain text bytes buffered. The data was originally sent encrypted by the server. W24 decrypted the cipher text in the background.
AT+iSRCV:0	Command to retrieve buffered plain text.
<i>I/1267:<...1267 bytes of plaintext data...></i>	W24 transmits buffered data to host.
AT+iscls:0	Close socket handle 0.
<i>I/OK</i>	SSL3/TLS1 socket is closed.
<i>I/DONE</i>	W24 is offline.

Secure FTP Session on W24

W24 supports a secure FTP session using SSL3/TLS1 sockets for both the FTP command and FTP data channels. The command used for opening a secure FTP session is AT+iFOPS.

Secure FTP implementation in W24 is based on RFC 2228 (FTP security extensions) and the IETF Internet draft "Securing FTP with TLS" (draft-murray-auth-ftp-ssl-16.txt).

When the AT+iFOPS command is used to initiate a secure FTP session, W24 performs the following operations:

1. Opens an FTP control socket.
2. Sends AUTH TLS.
3. Performs the SSL3/TLS1 handshake.
4. Sends USER command.
5. Sends PASS command.
6. Sends PBSZ 0, followed by PROT P.

Once the data channel TCP socket is established, all subsequent data connections (send or retrieve files as well as directory listings) start with an SSL3/TLS1 handshake. When a data socket is re-opened for another FTP command, W24 attempts a quick re-negotiation using the previous SSL3/TLS1 session parameters.

+iSSL - Secure Socket Connection Handshake

Syntax: AT+iSSL:<hn>

Negotiates a secure SSL3/TLS1 connection over an open TCP/IP socket.

Parameters: <hn> = A previously open TCP/IP socket handle.

Command Options:

<hn> Must be obtained using the AT+iSTCP command during the current Internet mode session. Or a socket **accepted** by a listening socket.
When a Network Time Server is defined and NTOD is set to 1, W24 confirms the server's certificate date validity using the retrieved network time. If, for some reason, the network time is not retrieved successfully, W24 does not accept the certificate until the time is retrieved successfully.

Result Code:

I/OK If the SSL3/TLS1 negotiation is successful.

I/ERROR Otherwise.

+i[@]FOPS - Secure FTP Open Session

Syntax: AT+i[@]FOPS:<server>[,<port>]:<user>,<pass>[,<acct>]

Opens a secure FTP link to a secure FTP server.

Parameters:

<server> Logical name of the FTP server or the server's IP address.

<port> Optional FTP port in the range 0-65535.

<user> FTP user's name.

<pass> FTP user's password.

<acct> Optional FTP account.

Command Options:

<server> The server name may be any legal Internet server name that can be resolved by W24's Domain Name Server (DNS) settings. The server name may also be specified as an absolute IP address given in DOT form.

<port> Specifies the FTP server's listening port. If not specified, port 21 (decimal) is assumed.

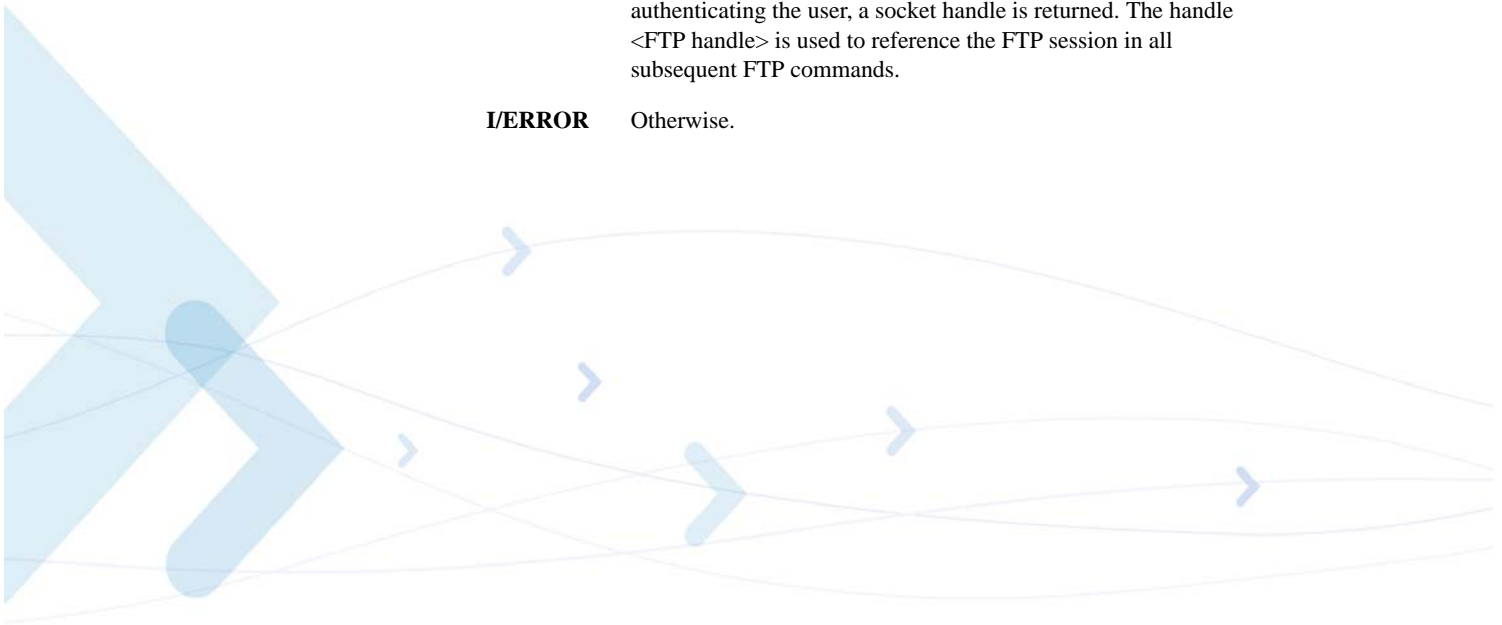
<user> User's name string. This must be a registered user on the FTP server. Some servers allow anonymous login, in which case *user*=anonymous.

- <pass>** Password for user authentication. If special characters are used, the password must be specified within quotes. It is customary that servers that allow anonymous login request an e-mail address as a password.
- <acct>** Some FTP servers require an account in order to allow a certain subset of the commands. In this case, the account name must be specified when opening the FTP link.
- @** The optional @ is used to flag the Force PASV mode. When @ is specified, W24 uses only the PASV method when opening a data socket to server for FTP data transfer.

Result Code:

- I/<FTP handle>** Upon successfully connecting to the FTP server and authenticating the user, a socket handle is returned. The handle <FTP handle> is used to reference the FTP session in all subsequent FTP commands.

- I/ERROR** Otherwise.

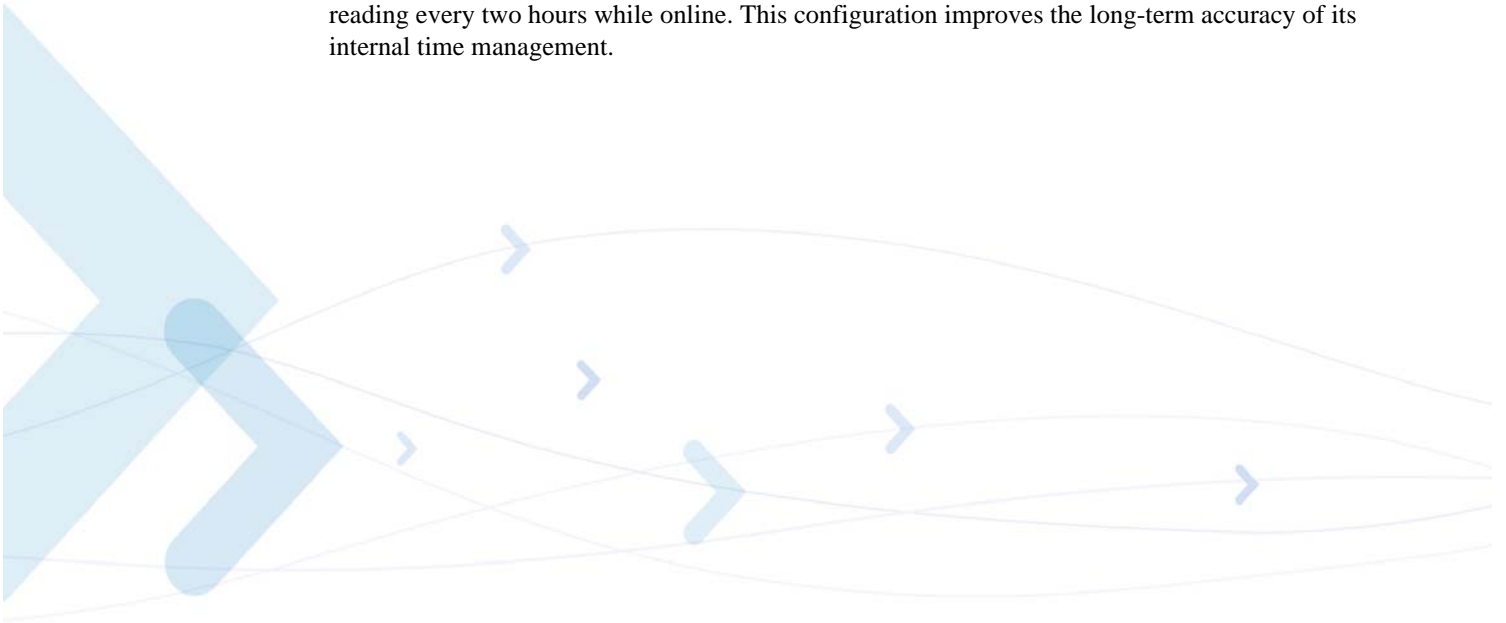


Network Time Client

W24 incorporates a Simple Network Time Protocol (SNTP) client. With this protocol support, W24 can be configured to check SNTP servers for current time and date each time it goes online. W24 is configured to retrieve time data from a Network Time Server each time it goes online with the NTOD parameter. After updating its internal Time-Of-Day (TOD) registers at least once, W24 continues to keep track of time independently, even after it goes offline.

When W24 contains real TOD data, e-mails sent are automatically stamped with Time and Date of delivery, according to RFC (822) definition for the date header field. In addition, the AT+iRP8 report returns the current time and date.

W24 also contains parameters to configure local GMT offset and a DST (Daylight Savings Time) rule. These parameters allow W24 to determine the local TOD. When W24 is configured for TOD retrieval from a Network Time Server, W24 automatically retrieves an updated time reading every two hours while online. This configuration improves the long-term accuracy of its internal time management.



MIME Encapsulated E-Mail Messages

W24-Generated Binary Message Formats

Binary e-mail messages are sent via W24 using one or more AT+iEMB commands. The message format is limited to an optional body of text and a single attachment.

The following fields are added by W24 to the main message header:

X-Mailer: iChip <software version>

Message-ID: <Unique #>@iChip

Mime-Version: 1.0

Content-Type: multipart/mixed; boundary="CONE-iChip-<software version>"

The message's preface contains the following text:

"This MIME message was coded by iChip."

If the host application includes a text body for the message, it also contains the following lines in its header:

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

X-iCoverpage: Email

When no textual body contents are included - this section is omitted.

The binary attachment section follows, beginning with a MIME attachment header containing the following fields:

Content-Type: <User defined media type>/<User defined media subtype>;

name=<User defined attachment filename>

Content-Transfer-Encoding: base64

where,

- <media type> := "text" / "image" / "audio" / "video" / "application"
- <media subtype> := <A publicly-defined extension token.>
- <filename> := <User-defined name (including extension)> or <unique filename>
- <media type> defaults to "application" when otherwise not defined.
- <media subtype> defaults to "octet-stream" when otherwise not defined.

Following the header, a base 64-encoded data stream includes the entire binary data transferred to W24 from the host.

MIME-Related AT+i Commands and Parameters

Binary images are transferred to W24 for MIME message encapsulation via one or more AT+iEMB commands. An AT+iEMB command sequence must be terminated by the AT+iE* command, indicating the end of the binary e-mail message.

When several consecutive AT+iEMB commands are used, the host must issue the commands with an inter-command delay, which does not violate the SMTP server's timeout constraints. Otherwise, the SMTP server will timeout and abort the session. Average SMTP servers allow for delays in the range of 30 to 120 seconds. Additional AT+i commands may be interlaced within a sequence of AT+iEMB commands, except for the following AT+i commands: AT+iEMA, AT+iRML, AT+iRMH, AT+iRMM, AT+iRFU, AT+iRLNK, AT+iBDRA, and AT+iSNMD.

W24 does not limit the size of the binary attachment. However, ISPs do have limitations. An Internet connection is initiated immediately after the first AT+iEMB command, while the rest of the command is received. Once the connection to the SMTP server has been established, W24 acts as a pipeline, receiving binary info from the host, encoding it, and transmitting it to the Internet on-the-fly. Following the AT+iE* command, the e-mail is terminated and the Internet connection closed.

The escape sequence command (+++) is allowed within an AT+iEMB command, provided there is a half-second silence period before the (+++) is sent. Upon receiving the escape sequence, W24 aborts and orderly closes the Internet session. The partial mail message is not sent to the destination.

Binary Attachment Parameters

Table 2-3 shows the binary attachment parameters.

Table 2-3: Binary Attachment Parameters

Parameter	Default	Description
MT	4 (application)	Media Type: 0 - Text; 1 - Image ; 2 - Audio ; 3 - Video ; 4 - Application
MST	octet-stream	Media Subtype String. For a list, see “MIME Content Types and Subtypes” on page A-1 .
FN	None	Attachment File Name (inc. extension). If a file name is not defined, W24 generates a unique filename without an extension.
BDY	None	ASCII text to be included in the e-mail's body in addition to the attachment. (Multiple lines allowed).

Defining a Textual Body for Binary Messages

1. Permanent textual body contents:

```
AT+iBDY:<text lines> ... <CR>.<CR>
```

The maximum fixed body size allowed is 96 characters (including embedded <CR><LF>). The text body is included in all future binary messages. In addition, the textual contents are committed to non-volatile memory on board the W24.

2. Single session textual body contents:

```
AT+iBDY~<text lines> ... <CR>.<CR>
```

The maximum temporary body size allowed is 1K characters (including embedded <CR><LF>).
The text body is included in the next session binary message and then purged.

MIME-Encapsulated E-Mail Message Format

Note: Bold lines are added by W24.

Received: from JFK by FTGate SmartPop;

Tue, **23 Nov** 1999 09:26:21 +0200

Received: from mail.inter.net.il (hrz-153-147.access.net.il
[212.68.153.147])

by mail.inter.net.il (8.9.3/8.8.6/PA) with SMTP id OAA11594;

Mon, 22 Nov 1999 14:18:03 +0200 (IST)

Date: Mon, 22 Nov 1999 14:18:03 +0200 (IST)

From: m2m@motorola.com

To: gadyl@netvision.net.il

X-Mailer: iChip ic401d05

X-Serial: 123456

Return-Receipt-To: m2m@motorola.com

Message-ID: <15322@iChip>

Subject: iChip binary message via iModem

Mime-Version: 1.0

Content-Type: multipart/mixed; boundary="CONE-iChip-ic401d05"

X-UIDL: ad0c01ac458208bedea8b8522012e4b6

This MIME message was coded by iChip.

--CONE-iChip-ic401d05

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

X-Coverpage: Email

.

<Textual body, here>

.

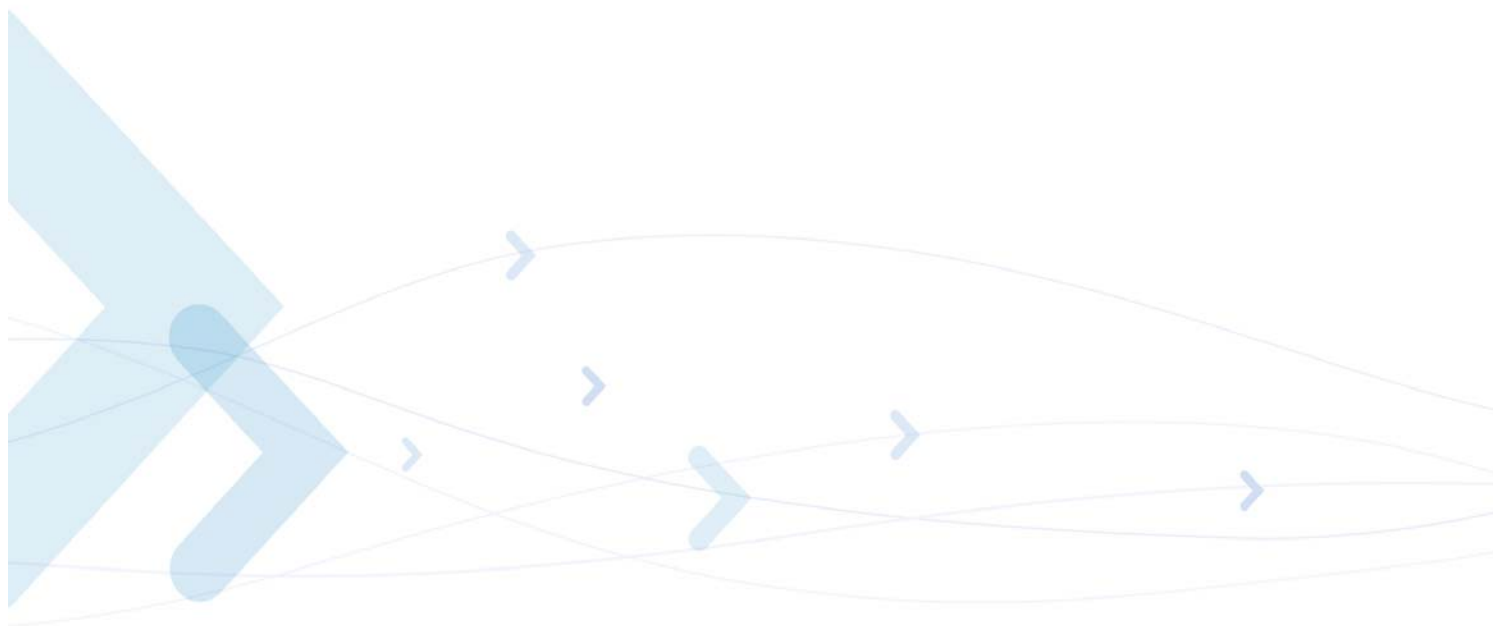
.

--CONE-iChip-ic401d05

Content-Type: image/tiff; name="FaxImage.tif"

Content-Transfer-Encoding: base64

```
.  
.   
.   
<Binary Base64-encoded data, here>  
.   
.   
.   
--CONE-iChip-ic401d05
```



Flow Control

Host -> W24 Software Flow Control

When issuing an AT+iEMB command to generate a binary e-mail, an AT+iSSND command to transfer data to a socket, an AT+iTBSN to send a binary stream to a Telnet server, or an AT+iFSND command to transfer a file, the host transfers a binary data stream to W24. At times, this stream may be very large.

Once W24 establishes a connection, it acts as a pipeline, transferring data received from the host to the Internet. However, the data rates at the host and Internet ends are not always balanced. This happens for several reasons:

- While W24 logs onto the Internet and establishes a connection, the host proceeds to send its data stream to W24. During this time W24 receives data from the host, but cannot send it out.
- When sending MIME attachments, W24 encodes the binary data using base 64. This roughly inflates binary data by 30%. Thus, more data needs to be transmitted than is received from the host.
- When using a TCP/IP socket, W24 might need to re-transmit packets.

The amount of buffer space available in the W24 to accommodate for this imbalance is limited. Therefore, a flow control scheme is required to regulate host<->W24 communications. The FLW parameter is set to reflect the preferred flow control mode.

The software-driven flow control protocol is defined as follows:

1. While the host is transferring the binary stream, following the +iEMB, +iSSND, or +FSND prefixes, W24 issues a 'WAIT' control character when it needs to pause the host. The host application is required to monitor its serial receive line and pause the transmission when a 'WAIT' control character is received.
2. To resume the host transmission, W24 issues a 'CONTINUE' control character. The host is required to monitor its receive line after being paused in anticipation of this control character. Once received, the host might continue to transfer the data stream.
3. If an error occurs during the Internet session while the host is transferring the data stream (or while paused), W24 issues an 'ERROR' control character if some error occurred. Immediately after issuing this control character, W24 aborts the Internet session and issues an 'I/ERROR (error number)' string. The host must cease transmitting the data stream when the 'ERROR' control character is received.

The control characters are given in [Table 2-4](#).

Table 2-4: Software Flow Control Characters

Control	ASCII Dec	ASCII Hex	Mnemonic
WAIT	22	0x16	SYN
CONTINUE	24	0x18	CAN
ERROR	5	0x5	ENQ

Software Flow Control Diagram in Binary E-Mail Send

See [Figure 2-4](#).

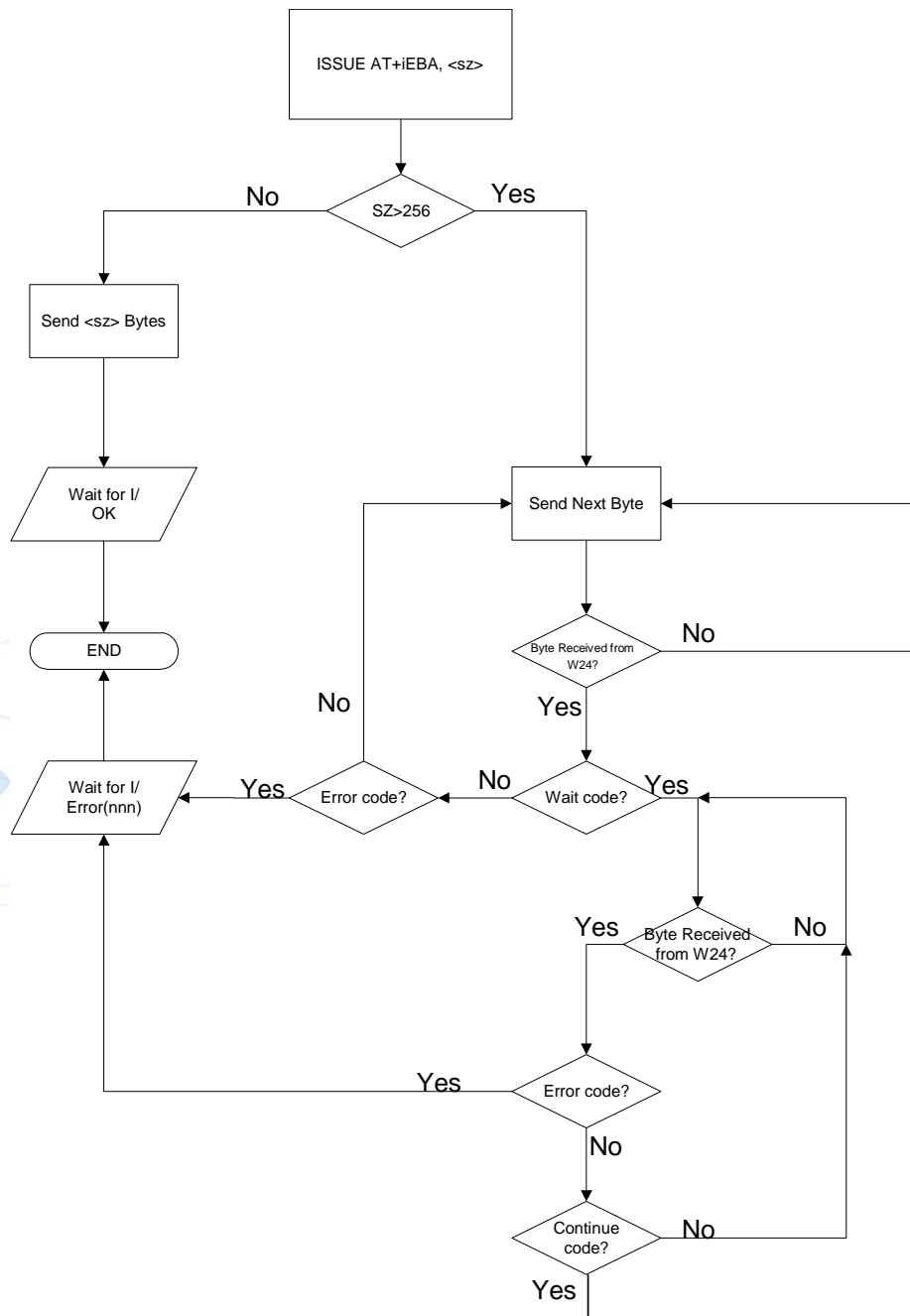


Figure 2-4: Software Flow Control in Binary E-Mail Send

Software Flow Control During a Socket Send

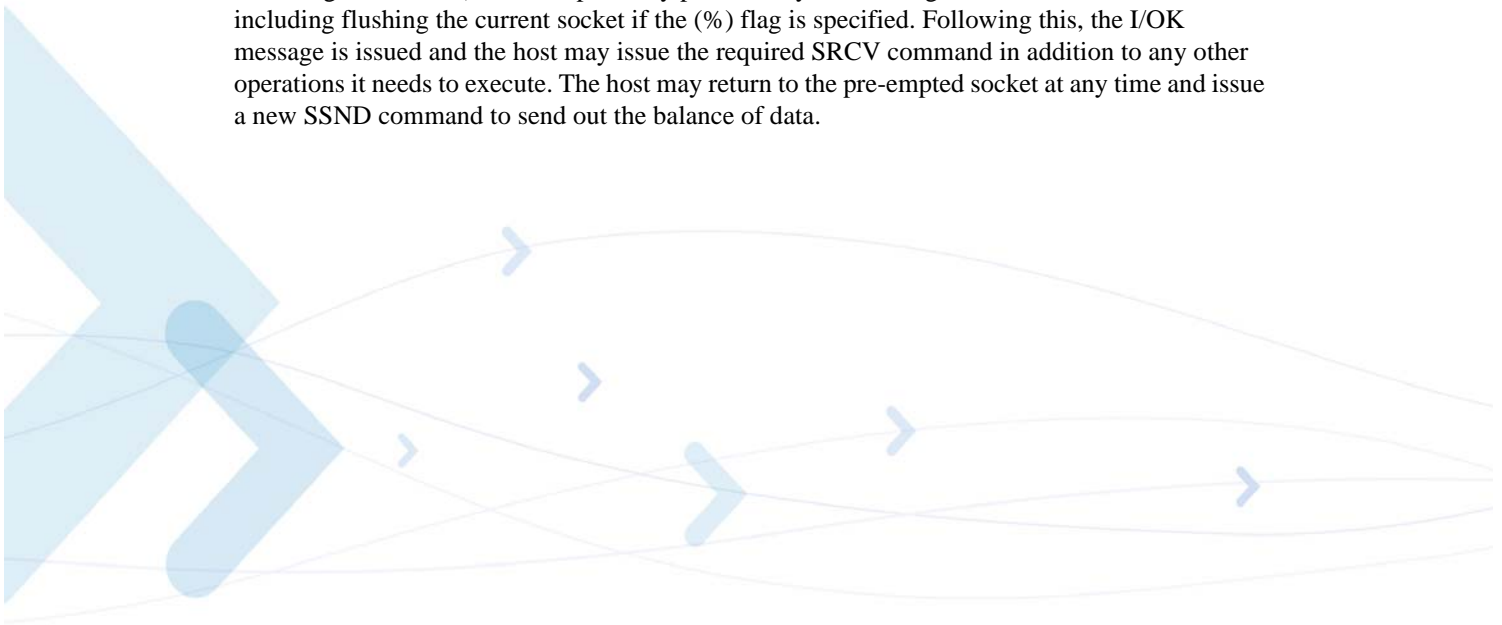
When a WAIT control is sent to the host during a socket send (AT+iSSND) command, it is automatically followed by an RP4 socket status report in the following syntax:

```
I/( <sock0sz>, <sock1sz>, ... , <sock9sz> ) <CR/LF>
```

See the AT+iRP command for a full description.

While the host is waiting for the CONTINUE control, it may analyze the sockets' input buffer status. If the host detects a need to execute a socket receive command to empty one or more socket input buffers, it may escape the current SSND command by issuing a *'Pause'* sequence immediately after receiving the *'CONTINUE'* control.

The *'Pause'* sequence is defined as: half a second of silence followed by (---) (three consecutive minus sign characters). W24 responds by prematurely terminating the SSND command, including flushing the current socket if the (%) flag is specified. Following this, the I/OK message is issued and the host may issue the required SRCV command in addition to any other operations it needs to execute. The host may return to the pre-empted socket at any time and issue a new SSND command to send out the balance of data.



Software Flow Control Diagram in Socket Send

See [Figure 2-5](#).

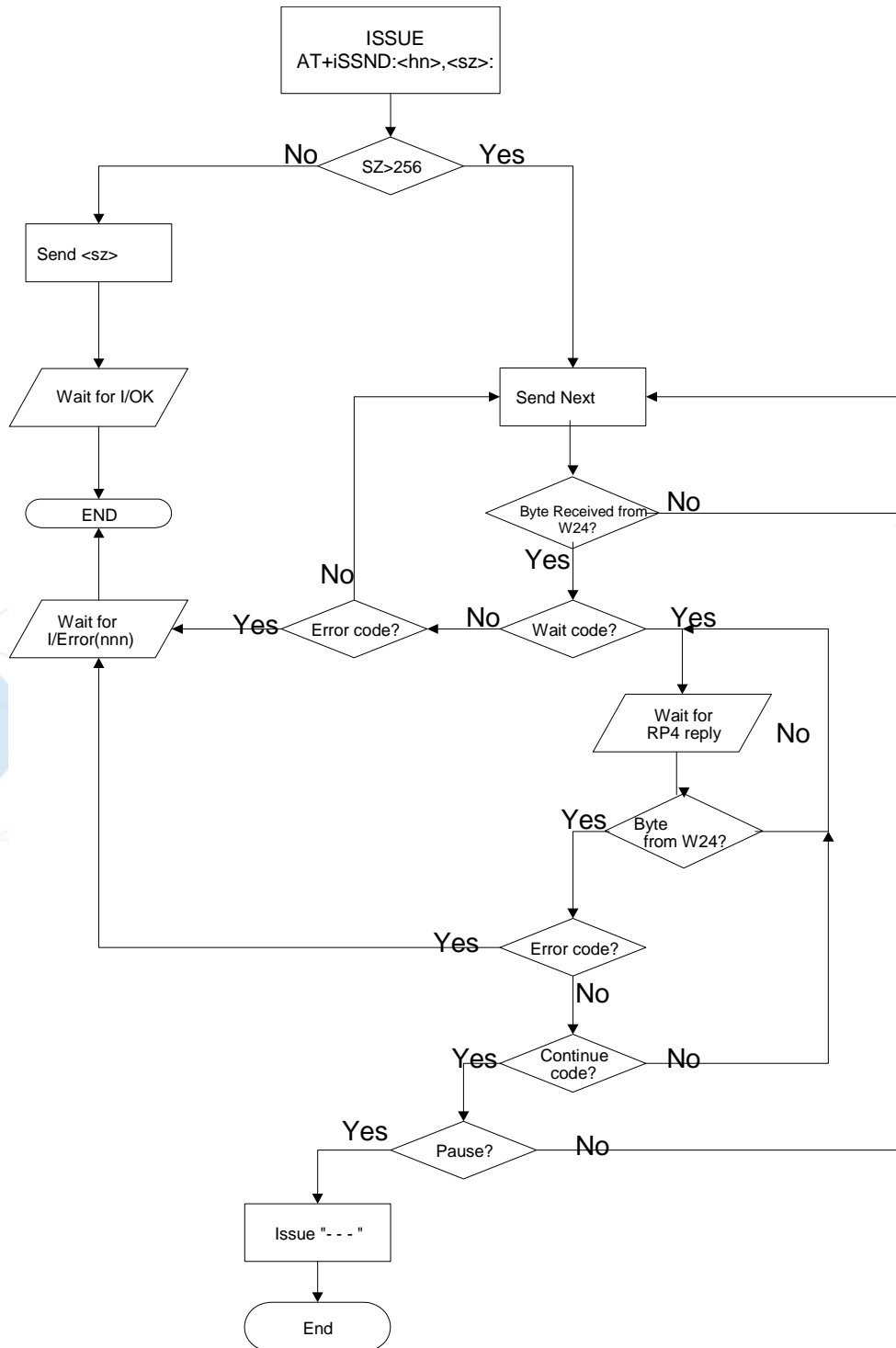


Figure 2-5: Software Flow Control in Socket Send

Host -> W24 Hardware Flow Control

As an alternative to the software flow control method, which requires some software attention on behalf of the host, W24 offers a hardware flow control mode.

This mode is selected by setting W24's FLW parameter Bit 0, using the AT+iFLW command. Note that to set FLW Bit 0, the \sim CTSH signal needs to be LOW (enabled), otherwise W24 returns I/ERROR (063). This convention safeguards W24 from lockup, which may arise if FLW Bit 0 is set while the \sim CTSH signal is constantly HIGH.

For hardware flow control to operate properly, the \sim CTS and \sim RTS signals between the host and W24 UARTS must be interconnected (see [Figure 2-6](#)).

The W24 \sim CTSH and \sim RTSH signals can be shorted to circumvent hardware flow control.

Under this mode, W24 assumes that the host transmission might be paused by de-asserting the \sim CTS signal. The host must adhere to this convention. Most UARTs support hardware flow control. However, if this is not the case, W24's \sim CTS signal must be monitored by the host software on a general purpose I/O.

The host can also pause W24 by de-asserting its \sim CTS signal.

If a transmission error occurs during processing of a send command (EMB, SSND, TBSN, FSND), W24 accepts all remaining characters pertaining to the current command (as specified by the <sz> parameter) before returning the relevant I/ERROR response.

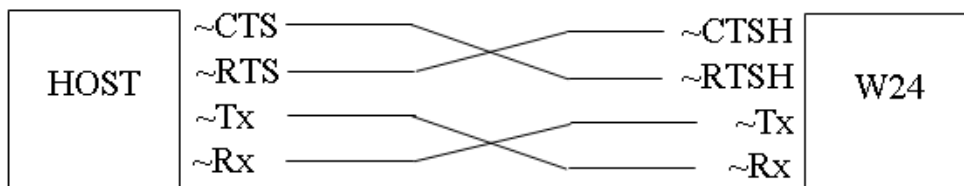


Figure 2-6: Minimum Hardware Flow Control Connections

Remote Firmware Update

Introduction

W24 accepts remote firmware updates from an HTTP or FTP server. The firmware update is stored as an .imz file on the host server and downloaded by W24 acting as a client. W24 replaces its existing firmware with the new one through a special application that is part of the .imz file. This method is especially convenient when managing firmware updates in a globally distributed install base of internet-enabled devices.

Updating Firmware from a Remote Server

This method involves placing the firmware update .imz file on an HTTP or FTP server. W24 has the provisions to use its respective HTTP or FTP client to download the firmware update file and perform the update process.

Before the actual remote firmware update command can be issued, the following parameters must be set:

- USRV - Defines the protocol to be used (HTTP or FTP), and the name of the host on which one or more .imz files are stored.
- UUSR - Defines FTP user name (FTP only).
- UPWD - Defines FTP user password (FTP only).
- UEN - This flag indicates whether W24 updates to a firmware version that is newer than the currently installed one only, or to any firmware version it finds.

In addition, an appropriate .imz firmware update file must be placed on the remote server at the location specified by the USRV parameter.

Once the above parameters are defined, the firmware update process can be initiated by sending the following command to W24:

AT+iRFU

W24 returns **I/OK** to acknowledge receipt of the command. As the update process may take up to 4 minutes to complete, W24 issues an **I/UPDATE** message to notify the host that it is in the process of updating its firmware. The host must allow for an extended delay period until W24 completes the process. Once completed, W24 re-boots the new firmware and issues an **I/DONE** message when in dialup mode, or an **I/ONLINE** in WLAN mode.

Several safeguards have been instated to ensure a successful firmware update. The firmware update file is structured in a specific format, which allows W24 to authenticate its origin as a legal firmware image. W24 also verifies that the firmware update is the correct version for its hardware environment. W24 rejects an update file if it contains an image that is identical to the one already installed.

The remote firmware update procedure is detailed below:

1. W24 downloads the new firmware imz file.
2. If the download fails, W24 returns an error message and continues to work as before.
3. If during the download W24 is going over a reset cycle (SW or HW), W24 re-boots and executes the old firmware.
4. If the download is successful, W24 authenticates the firmware image file.

5. W24 replaces the old image with the new image.
6. If the replacement process fails, for example due to power failure, W24 re-boots from boot loader in the flash memory and re-tries the replacement process until successful.
7. If the replacement process is successful, W24 re-boots and executes the new firmware.

+iRFU - Remote Firmware Update

Syntax: AT+iRFU

Downloads and updates W24 firmware from a remote HTTP or FTP server. The value of the USRV parameter is used to determine the remote server from which to download the firmware. The value of the UEN flag is used to determine whether to update any firmware version or only a version that is newer than the one already installed. In addition, if an FTP server is specified for download, the UUSR and UPWD parameter values are used to determine FTP user name and password.

Result Code:

I/OK To acknowledge successful receipt of the command.

I/ERROR Otherwise.

Followed by:

I/UPDATE If a qualifying firmware update .imz file is found.

I/ERROR Otherwise.

Followed by:

I/DONE After successfully updating new firmware in dialup mode.

I/ONLINE After successfully updating new firmware in WLAN mode.

I/ERROR Otherwise.

W24 Parameter Update

Introduction

The W24 remote parameter update file allows users to remotely modify various non-volatile parameters in W24 products. The file is an ASCII-formatted text file, edited by the user or created by a dedicated application. The file's size must not exceed 10k.

The remote parameter file (RPF) naming convention is *<filename>.rpf*. If a parameter is assigned a legal value within the file, that value replaces the current value in W24's non-volatile parameter database. A parameter value that is not referred to in the file, or that is not defined using the correct syntax rules, specified below, does not affect the current parameter value.

Remote Parameter File (RPF) Structure

The RPF file must include the letters "RP_" as its first 3 characters, and can include additional header lines (defined below), as well as various parameter assignments. Assignments follow the rules defined for parameter settings, but excluding the AT+i prefix. For example, to assign the value myname to the POP3 mailbox name parameter, the correct assignment is MBX=myname. This is equivalent to the host sending AT+iMBX=myname to W24. Each line, terminated with <CR>/<LF>, can contain one assignment only. The order of assignments is not important, except for the RPF header parameters, which must be first and must follow the header definitions below. After the first non-RPF header parameter, additional header parameters are ignored.

Comment lines can appear anywhere in the file. Comment line syntax is defined as:

```
#<anything>CR/LF
```

The first line in the file that is not a comment line is considered the authentication header line and must have the following syntax:

```
RP_[GROUP=<string><space_character>][RP_DEST=<string>]CR/LF
```

The remainder of the header must contain lines with the following syntax:

```
<header_parameter_name>=<general_parameter_value>CR/LF
```

Header Parameter Names and Values

Table 2-5 shows the header parameter names and values.

Table 2-5: Header Parameter Names and Values

Name	Value	Default
RP_DEST	Single string, no space characters.	NONE
RP_GROUP		NONE
RP_START_FROM_FACTORY_DEFAULTS	YES/NO	NO

- **RP_GROUP** - If the RPF Group Name parameter contains a value, the RPF file must include an RP_GROUP definition and its value must be identical to the RPF value. Otherwise, the parameter update file will be rejected. Nevertheless, if the RPF parameter is set to the special value (*) (match any), the RPF file will be accepted with any value of RP_GROUP, as well as without any value at all. The RPF Group Name parameter can be viewed and changed by sending an AT+iRPG? command to W24.
- **RP_DEST** - If the RPF file contains this parameter, the parameter update file will be rejected unless the value given in this parameter is identical to the unique ID of the W24 it was sent to. The unique ID can be viewed by sending an AT+iRP5 command to W24, but cannot be changed. This feature facilitates sending a parameter update to a specific W24 controller only.
- **RP_START_FROM_FACTORY_DEFAULTS** - This flag defines the initial value of parameters. A YES value will initially restore all W24 parameters to their factory default values before processing the new RPF file values.

Uploading a Parameters Update File to W24

By default, receiving and processing a parameters update file is disabled in the W24. To enable this option, the RPG parameter must be set to some value. If a value other than (*) is set, the value must match the parameters update file RP_GROUP value. This feature facilitates group updates, and can be used as a password to secure parameter updates.

A remote parameters update file can be uploaded to W24 using W24's internal configuration site.

The nonvolatile parameter RPG controls the parameter update. If it does not contain a value, the update process is effectively disabled. If it contains an (*), it is fully enabled. If it contains a value, the update process is restricted to RPF files containing that value in the RP_GROUP header parameter.

Note: See [“Sample Parameter Update File”](#) for a sample RPF file.

W24 Embedded Web Server

Introduction

W24 includes a web server that handles HTTP 1.0/1.1 web interactions independently of its host processor. It allows system designers to build web-based products, which can be remotely monitored, configured, and managed via the Internet using a standard web browser interface.

W24 devices host two on-chip websites stored in non-volatile memory. One website is inherent to the W24 firmware and dedicated to W24 configuration and maintenance. The second site is uploaded to W24 for device application use. This website can include multiple linked HTML pages, links to external pages, images, graphics, Java applets, WAP pages, and more. A special facility allows the web pages to include references to the embedded application's variables.

W24's embedded web server is designed to integrate with the existing W24-to-host API methodology based on AT+i command interface.

Features

- Responds to standard web browser GET and POST commands issued on port 80.
- Supports up to three concurrent remote browsers.
- Serves on-chip HTML pages stored in non-volatile memory.
- Can incorporate WAP pages to allow browsing W24's website using an Internet-enabled cellular handset.
- The internal W24 configuration website supports remote W24 parameter configuration, remote W24 firmware upload, and remote application website upload. This is achieved using a standard web browser. Configuration access is protected by an SHA1-encrypted password mechanism.
- Supports monitoring and controlling the host device using a pre-defined set of parameters embedded within the application website (also SHA1 password protected).
- Allows OEMs to design their own embedded website using standard web authoring tools along with a windows-based website packing utility.

Web Server Modes

Two web server modes are defined as (see [Figure 2-7](#)):

- W24 configuration mode
- Host interaction mode

Each of these modes is supported by a dedicated website and a parameter access password.

The W24 configuration mode allows remote W24 configuration. It encompasses web interactions between W24 and a remote browser to carry out W24 parameter maintenance and W24 firmware and application website uploads. The host processor does not take part in the interactions under this mode. Moreover, the host processor is not required at all for this mode to operate. Once a W24 is online and in possession of an IP address, any remote browser may surf to the W24 and update its non-volatile parameters without the host's involvement. The W24 configuration site is located at:

[HTTP://<W24_IP_Address>/w24/](http://<W24_IP_Address>/w24/)

In Host interaction mode, W24 is used to host, serve, and manage web interactions with a remote web browser on behalf of the embedded device's host processor. The host gains access to the web-based parameters via AT+i commands sent to W24 through the serial connection..

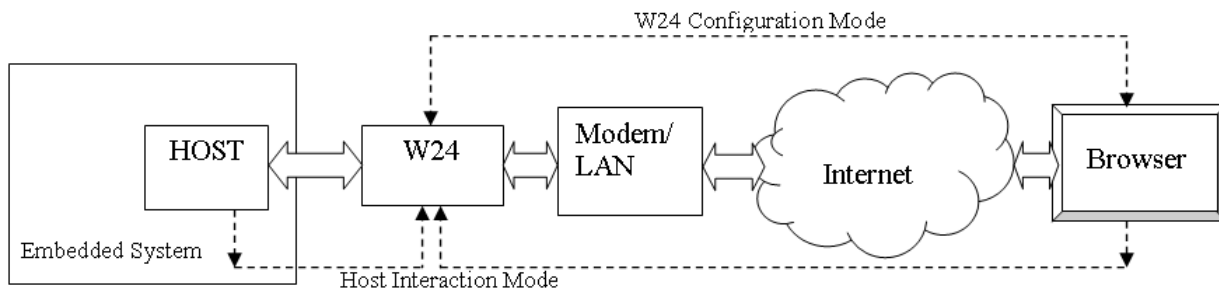


Figure 2-7: W24 Web Server Modes

The Application Website

The application website is stored in non-volatile memory. It consists HTML code, which can include links to local or remote web pages, graphic images, text files, Java applets, WAP pages, and more.

Device manufacturers can design their own embedded website using any web authoring tool. The W24 implementation supports a maximum website size of 64K. The site is uploaded to W24 through the serial connection, or through W24's configuration website.

Parameter Tags

W24 and host real-time parameters can be referred to in the embedded websites through the use of Parameter Tags. When Parameter Tags are placed in an HTML web page, actual values are sent by W24's web server component when the page is served out. Parameter Tags are also used to change corresponding parameter values from a remote web browser. Syntactically, Parameter Tags are parameter names enclosed between two (~) characters. If the (~) character needs to be included in a Web page, two consecutive (~) characters must be used (~~).

The W24 Internet configuration parameters defined in the AT+i API retain their name when used as Parameter Tags. For example, the value of the *TOA* AT+i parameter (Send to E-Mail Address) may be referenced in the website by *~TOA~*.

Host Parameter Tags defined by the parameter name *<param>*, may be referenced in the website using *~<param>~*. *<param>* can be any freeform parameter name consisting of a single word that does not include blanks or W24 delimiters. For example, a parameter reflecting a temperature reading can be called *temperature* and referenced in the website as *~temperature~*.

W24 Configuration Mode

W24 configuration entails monitoring and updating W24 parameter values. By making use of W24's inherent configuration website, a W24 device can be configured remotely using a standard web browser in addition to being configurable locally using the Ymodem protocol over the serial link, via PSTN in a modem environment, or remotely via e-mail. The W24 RPG parameter is used to password-protect remote W24 parameter updates. See Security and Restrictions.

The configuration site includes web forms to monitor and update most W24 parameters and an upload page consisting of file upload forms. Note that, the following W24 parameters cannot be configured remotely and are therefore not displayed on W24's configuration website:

- WiFi security parameters
- Fast USART parameter (BDRD)
- Analog-to-digital converter (ADC) parameters

Each upload form allows file uploading using the POST method for a single file. The forms support uploading the following files:

- Firmware update *.imz file
- Parameters update *.rpf file
- Packed application website *.img file

When new firmware (*.imz file) is uploaded to W24, W24 submits an acknowledgment page to the browser, after receiving the complete *.imz file, and then goes offline and updates its firmware.

In some rare cases, W24's internal configuration website may be accidentally corrupted. This happens when W24 fails to complete a remote firmware update process via web. To resolve this problem, W24 includes a recovery website. This website allows a user at the remote browser end to upload the .imz file again in order to restore W24's internal website.

W24's configuration site is located at:

HTTP://<W24_IP_Address>/w24/

Host Interaction Mode

Host Interaction mode allows OEMs to design and implement a product-related embedded website that is managed by W24 on behalf of the host. The host-defined embedded website supports live host parameter monitoring and updating by a remote browser. This is achieved by a dynamic AT+i layer implemented across the serial link between the host and W24.

The application developer creates a website using conventional web authoring tools. The HTML or WAP files can then be edited to contain Parameter Tags. Parameter tags are regarded as placeholders in HTML or WAP files. They are replaced on-the-fly with real-time values as the page is served to the browser. Browsers may also change values of Parameter Tags in order to submit the value back to the host via W24. This is done by defining the Parameter Tag in the NAME field in an HTML FORM (without the (~) characters). The W24 WPWD parameter is used to password-protect remote Parameter Tags update. See Security and Restrictions.

Once a website is created and Parameter Tags are edited in, the site is packed and uploaded to W24. The website is linked into the W24 firmware, automatically expanding the existing AT+i command set to encompass the website Parameter Tags. This happens when the web server is activated using the +iWWW command.

Extended AT+i commands have the following syntax:

```
> AT+i<param>=<value>
```

```
> AT+i<param>?
```

for setting and querying Parameter Tag values, respectively.

For example, the ~temperature~ Parameter Tag referenced in a web page, can be set using:

```
> AT+itemperature='45 Deg.'
```


and queried using:

```
> AT+itemperature?
```

When the host issues a Set Parameter Tag Value command, W24 links the updated value to the Parameter Tag and stores it in its internal RAM. In response to a browser's GET request, the real value is substituted everywhere in the page where the Parameter Tag exists while the page is being served, on-the-fly.

Parameter Tag values are printable ASCII text. This convention allows implementing any part of an HTML or WAP page as a parameter tag: numeric values, links, file names, HTML code, etc. A Parameter Tag value is limited to 256 characters.

Parameter Tag values can be changed and submitted from the browser end using HTML forms. W24 stores the updated values and responds appropriately to host AT+i parameter query commands. Thus, the host can poll specific parameters for value changes. Status Report 7 (AT+IRP7) can be used to facilitate polling on all application web parameters. RP7 returns a bitmap result, where bit 10 is set to '1' if one or more application web parameters have been remotely changed. The W24 DATA_RDY signal is an associated hardware signal that can be used to generate an interrupt on the host CPU when new data has been buffered in W24. The ISR can issue an RP7 to determine if the new data is a result of an application web parameter change.

The AT+iWNXT command can be issued to scan through the application web parameters that have been remotely updated and not yet retrieved by the application.

The W24 application site is located at:

[HTTP://<W24_IP_Address>/](http://<W24_IP_Address>/)

Website Creation, Packing, and Uploading

Device manufacturers can design their own embedded website using any typical web authoring tool. A website can include one or more files residing in a dedicated file directory structure on the designer's PC. The topmost directory of this structure is referred to as the website root directory. The root directory must contain an HTML page named index.htm, which serves as the default home page.

Before downloading the website to a W24 device, the entire website needs to be packed. In order to pack the site into an uploadable image file, the designer must run the web packing utility and specify the root directory of the site. The utility packs all files in the root directory and its subfolders in a format suitable for W24. If the site contains Parameter Tags, the user is prompted to enter a maximum value length for each Parameter Tag. Any Parameter Tag specified with a zero length value will not be included in the resulting packed file. After the user has entered all parameters' max value length, the user is prompted to specify a destination for the packed file.

The following restrictions apply when creating the packed website:

- The length of a single Parameter Tag must not exceed 256 characters.
- The sum of all Parameter Tags' value lengths must not exceed 8K.
- The total packed file must not exceed 64K.

To take effect, the packed website file needs to be uploaded to W24. This is done through W24's configuration website over the Internet.

Manipulating Variables in the Application Website

The application website is composed of HTML or WAP files, which may contain links to internal or external websites, Java Scripts, VB scripts, graphic files, and more (See list of supported file types). Using Parameter Tags, the page can also be used to dynamically display and update values of W24's configuration parameters and device-specific Parameter Tags in the manner described above.

For example, to display the current value of the headline web parameter, enter `~headline~` anywhere on the page, as in the following example:

```
<HTML>

<HEAD>

<TITLE>SAMPLE PAGE</TITLE>

</HEAD>

<BODY>

<h1>~headline~</h1>

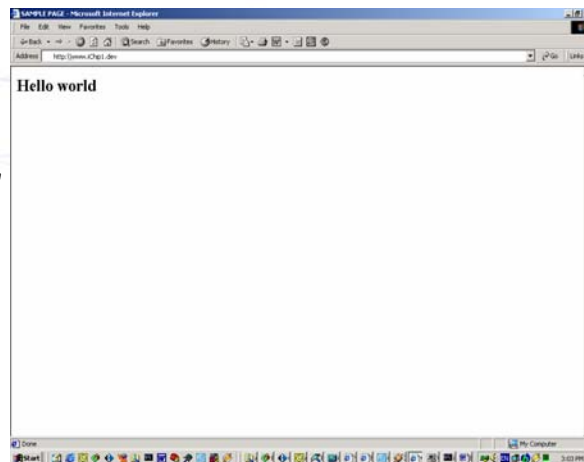
</BODY>

</HTML>
```

When serving this home page, W24's web server replaces the `~headline~` string in the served page with the current value of that parameter.

For example, if the host issues:

`AT+iheadline="Hello world"`



A browser pointing to W24's URL address will display the image as seen on the right.

To update W24 configuration parameters via the web page, simply use W24's parameter names (excluding the AT+i prefix) in an HTML form.

For example:

```
<HTML>

<HEAD>

<TITLE>SAMPLE PAGE</TITLE>

</HEAD>
```

```
<BODY>

<FORM METHOD='GET' ACTION=''>

Dial To:<INPUT type='text' name='ISP1' value='~ISP1~'>

<input type='submit' size='8' value='Submit'>

</FORM>

</BODY>

</HTML>
```

Note that the variable name is used in the NAME field, while ~<parameter name>~ is used to display the current value.

After activating SUBMIT, the browser issues a GET command to W24's web server that includes the parameter's name and the new value entered in the form. The page is then served to the browser again with the updated values.

In addition to specifying W24 configuration parameters and Parameter Tags, it is also possible to display W24 reports and W24's MAC address. For example:

```
<table>

    <tr>

<td width=250><b>MAC Address: ~MACA~ <b></td>

    </tr>

    <tr>

<td width=250><b>Bootblock Version: BBIC~RP3~</b> </td>

<td width=400><b>Firmware Version: ~RP1~</b></td>

    </tr>

    <tr>

<td width=250><b>Serial Number: ~RP5~ </b></td>

<td width=400><b>Hardware Version: ~RP0~</b></td>

    </tr>

</table>
```

Security and Restrictions

The authorization to view and update W24's configuration parameters, firmware, or application website via the web can be password-protected using the AT+iRPG parameter (Remote Parameter Group/Password).

When the RPG parameter in a W24 device contains a value, it is considered a password that restricts remote W24 parameter viewing/updates. By default, W24's configuration site can be viewed (browsed), unless the Security Disable Mode (SDM) bit 3 is set, in which case the user is authenticated by submitting the RPG value. To enable remote updates, a distant user is always authenticated by submitting that value. The W24 configuration site includes an authentication form that automatically pops up on the remote browser when parameter updates are attempted. The password submitted through this form must match the actual value of W24's local RPG parameter. Otherwise, remote value updates are rejected.

W24 uses the industry standard SHA1 algorithm to authenticate the remote user. According to SHA1, the password typed into the authentication form is not literally communicated back to W24. Rather, a SHA1-encrypted token is transferred. To achieve this, W24's web server sends a JavaScript, which calculates SHA1 encryption at the browser end together with the authentication form. W24 also issues a different random number, used as part of the encryption key, each time authentication is required, to eliminate the possibility of impersonation based on eavesdropping to a legal authentication session.

If the RPG parameter is empty (AT+iRPG=""), remote W24 configuration parameter update is fully restricted. In other words, it is not possible to update configuration parameter values using a remote browser. Conversely, if the RPG parameter contains an (*) character (match any), the configuration parameters can be updated freely, without requiring authentication at all.

The Parameter Tags defined in the application website are secured from remote updates in the same manner as the W24 configuration parameters. In this case, the authentication password is stored in W24's local parameter WPWD (Web Password). If the WPWD parameter contains a value, a remote user needs to issue this value as an authentication password in order to gain update access to the application level Parameter Tags. Like in the case of the RPG parameter, if WPWD is empty, application level Parameter Tags are fully restricted, whereas when WPWD contains an (*), updates are unrestricted and authentication is not required.

When authentication is required, W24's web server automatically issues an authentication form to the remote browser in response to an attempt to update Parameter Tags. This procedure allows the application site to include HTML submit instances anywhere in the website without worrying about the authentication process. Authentication is automatically activated depending on the local value of the WPWD parameter.

Authentication needs to be submitted only once per session in order to enable browsing, Parameter Tags, or W24 configuration updates. In addition, authentication automatically expires after 10 minutes of inactivity.

Parameter Update Error Handling

An attempt to assign an illegal value to a parameter will fail and a string containing the relevant error message will be stored in a special W24 Parameter Tag named WST (Web Server Status). This value can be displayed in the page as any other parameter value (using ~WST~). For Example:

```
<b>Update Error Message: ~WST~</b>
```

File Types Supported by W24's Web Server

- The following files can include parameter tags:
- .HTM, .HTML, .JS, .VBS, .INC, .STM, .XML, .XSL, .HTC, .CSS, .WML, .WMLS, .XHTML
- The following files cannot include parameter tags:
- .CLASS, .GIF, .JPG, .PDF, .DOC, .PPT, .BMP, .XLS, .WMLC, .WMLSC, .WBMP

W24 RAS Server


Introduction

W24 features an internal Remote Access Server (RAS) that allows a remote dialer to dial into W24 using an active modem platform. When configured as RAS, W24 answers the incoming call and negotiates a PPP connection.

W24's RAS supports acknowledging an IP address request from the remote dialer side, as well as assigning a default IP address. Once the connection is established, the client can browse W24's website. (If the AWS parameter is set to a non-zero value.) All other W24 IP protocol functionality is also enabled, allowing the host to issue Internet protocol AT+i commands based on the PPP connection. Note, however, that since W24 is not connected to an actual ISP in this mode, W24 does not have access to the public Internet and thus only direct connections between W24 and the connected PPP client are possible.

RAS Parameters

Three parameters govern the use of W24's RAS server:



RAU	<p>RAS Login User Name.</p> <p>The RAU parameter defines the allowable user name for login purposes when W24 answers an incoming call as a RAS. The remote dialer must specify the correct user name and matching password in order to successfully complete the PPP connection. This parameter must have a non-empty value for the RAS feature to be enabled. Otherwise, when RAU is empty, W24's RAS is effectively disabled. When RAU contains the special character (*), RAS is enabled but no authentication is required.</p>
RAP	<p>RAS Login Password.</p> <p>The remote dialer must provide the correct password in order to successfully complete the PPP connection. When the RAP parameter is empty or contains a (*), any password string is accepted, in effect nullifying the authentication process.</p>
RAR	<p>Number of RINGs before picking up the line.</p> <p>When the RAS feature is enabled, the RAR parameter defines the number of RINGs that must arrive before W24 picks up the line and transfers control to its RAS.</p>

RAS Theory of Operation

When a remote client dials into W24, the modem RING strings are transferred by W24 (which defaults to transparent mode) to the host. When the RAS feature is enabled (RAU contains a value), W24 picks up the line and negotiates a PPP connection by issuing the ATA (modem) command after RAR RING strings have been received.

If the host chooses to manage a direct (modem-to-modem) data connection, it can pick up the line before RAR RING strings have arrived by issuing the ATA modem command.

During RAS PPP negotiations, W24 will reply only to (+++) (escape sequence) and AT+iRPn commands. Specifically, W24 replies "Connecting as RAS" to the AT+iRP2 (W24 status) command. The escape sequence can be used to abort the RAS session at any time. The AT+iRP2 command is the only means for the host processor to determine that a PPP session is in progress. W24 manages the RAS protocol internally and does not transfer any information to the host. Any other commands received from the host are disregarded by W24.

Once the PPP connection has been fully negotiated and established, W24 responds to all AT+i commands as when it is online. Specifically, W24 replies "RAS Connected" to the AT+iRP2 command.

As part of the PPP negotiation, W24 assigns itself the default IP 192.168.0.1 and allocates 192.168.0.2 as the client IP. However, if the client requests a specific IP, W24 always grants the client's request and uses the client's IP minus 1 as its own IP.

The following restriction to the minus 1 rule applies: If the IP requested by the client minus 1 is an IP address that ends with 0x00 or 0x255 as the last nibble, W24 assigns itself with the client's IP plus 1 instead of minus 1. This is done to assure that the IP that W24 assigns itself never violates the rule that defines that a network or host IP segment may not be all binary 1's, nor all binary 0's.

After a RAS PPP connection is established, W24 automatically activates the internal web server, if the AWS parameter is set to a non-zero value. Thus, the remote client can browse W24's website.

Auto PPP RAS Mode

W24 allows combining RAS and direct modem-to-modem communication sessions. A special mode, named Auto PPP RAS, supports dialing into the W24 with a PPP dialer or a regular modem.

Auto PPP RAS mode is enabled by enabling RAS mode and adding a +100 offset to the RAR parameter, where [*<RAR>-100*] determines the number of RINGS after which W24 automatically picks up the line and negotiates a PPP connection. The host processor can instruct the modem to pick up the line beforehand by issuing the ATA (modem) command or by setting the modem to auto-answer after less than [*<RAR>-100*] RING strings. This is normally done in order to manage a direct modem-to-modem (non-PPP) communication session.

When W24 is in the Auto PPP RAS mode, it monitors the data stream following the modem CONNECT line. If the first character transmitted by the remote end is (~) (0x7E), W24 defers to PPP negotiation. The (~) is the last character transmitted to the host end to signal that W24 has taken over the negotiations. Upon this event, W24 continues to negotiate a PPP connection internally in a manner similar to the procedure that occurs when W24 picks up the line after receiving <RAR> RING strings. If, however, the first character received from the calling end after the CONNECT line is not a (~) (0x7E), W24 remains in Transparent mode, and a regular modem-to-modem data session takes place.

SerialNET Mode

The RAS can also be enabled while W24 is in SerialNet mode. In this case, however, the modem RING strings are not forwarded to the host serial port. Once the PPP connection is established, W24 proceeds to act as it would after receiving a RING event and creating a PPP connection to a remote RAS server. That is, a listening socket is established on the LPRT socket, available for a SerialNET connection. This provides an alternative means to wake-up a SerialNET server device.

Lost Carrier

When W24 is online as a result of a RAS connection and the carrier signal is lost (due to an error or due to the PPP client closing the connection), W24 checks if the host used the PPP connection (tried to open an Internet session) during the connection. If the host did not use the connection, or W24 was in SerialNET mode, W24 silently performs a software reset and no indication of the disconnection is given to the host. Otherwise, if the host did use the connection, W24 acts as if this is a regular session created by the host that was terminated with a lost carrier signal. The error code is returned to the host on the next command that requires the use of the connection and only then will a software reset be performed.

Restrictions

Modem RING strings are not detected while the baud rate between W24 and the host is not yet established. This means that in order to use the RAS feature, one of the following must apply:

- BDRF is set to a fixed value (3-9 or h).
- W24 is in SerialNET mode with its baud rate defined by the SNSI parameter.
- An a or A was previously received from the host serial port and W24 has determined the host's baud rate.

In addition, Modem RING strings are not detected when W24 is in Modem Command (MCM) mode.

SerialNET Theory of Operation

Introduction

W24's SerialNET mode extends a local asynchronous serial link to a TCP or UDP socket across a WLAN or Internet. Its main purpose is to allow simple devices, which normally interact over a serial line, to interact in a similar fashion across a network without requiring any changes in the device itself. In order to achieve this, SerialNET mode defines a set of associated operational parameters, which determine the nature of the desired network connection. When W24 is put in SerialNET mode, it acts as a router between the device's serial port and the network.

Devices that communicate with a terminal over a serial link fall into three major categories: Output only (i.e. printers), Input only (i.e. controllers) and interactive (bi-directional communications). The latter are subdivided further into clients and servers. Generally, clients initiate communications by sending service demands to a server, while servers respond to client demands.

SerialNET mode reacts differently to client or server devices. When a client device initiates communications, SerialNET mode must establish a network connection to a remote server before data may flow between the two systems. On the other hand, when a remote client needs to invoke a device, the remote client first contacts the W24 and SerialNET is invoked to create a communication flow to the local server device.

SerialNET mode includes components to handle both server and client local devices. The W24 under SerialNET mode routes full-duplex data between a networked terminal and both types of devices.

SerialNET Mode

SerialNET mode is established by first defining all related parameters using AT+i commands, followed by a special Enter SerialNET Mode AT+i command.

Once in SerialNET mode, no additional AT+i commands can be sent, as the host serial link will be dedicated to raw local-device data. In this mode, auto baud rate is also disabled, since it cannot be guaranteed that the device will issue an a or A as its first character. Thus, a predefined fixed baud rate must be specified before switching over to SerialNET mode.

SerialNET mode extends across power-down, since it is assumed that once acting in this mode, W24 is connected to an AT+i aware host.

SerialNET mode can be terminated by:

- Applying power to the W24 with the MSEL signal pulled low for less than 5 seconds.
- Pulling low the MSEL signal for more than 5 seconds during runtime.
- Issuing the ESC sequence, defined as a half second delay followed by (+++) (three (+) characters), through the serial port.

When one of these occurs, W24 reboots after terminating SerialNET mode. At this point W24 reverts to its normal operational mode and again responds to AT+i commands.

Server Devices

Server devices linger until approached by a remote client. The remote client must know W24's IP and listening port address in order to establish communications.

WLAN-based devices and dial-up devices linger differently.

A WLAN device is normally online and may thus have an associated listening (passive) socket ready to accept remote socket connections. While in SerialNET mode, W24 establishes a listening socket on the port defined in its LPRT parameter. A remote client terminal can connect to that port.

A dial-up device is normally offline and must be awakened to go online at a precise moment. Moreover, once it connects to the Internet, it usually receives a dynamic IP address. This address must be communicated in some way to the client device in order to establish a link across the Internet. W24 resolves these problems by supporting a wake-up call and automatically implementing one or more IP registration procedures. This allows a client to wake up a W24 in SerialNET mode and retrieve its dynamic IP address from a registration server.

The W24 or in dial-up mode is offline by default, but waits for a RING signal on the modem to trigger it into activity. In this case, the remote client device dials directly to the W24 and hangs up after two rings. When contacted, W24 (under SerialNET mode) waits for the RING to subside and then dials into its ISP and connects to the Internet. If the RRMA parameter contains an e-mail address, W24 registers its IP address using the Email registration method. W24 then listens on the LPRT port for a socket connection. The recipient of the e-mail can use the registered IP address and port to create a link to W24's SerialNET socket.

If the RRSV parameter contains a server name and port, W24 registers its IP address using the Socket registration method.

If the RRWS parameter contains a URL, W24 registers its IP address using the Web server registration method.

Once connected, W24 transfers all arriving data from the local device over the serial link. Device responses are routed back to the initiating client. Data flows freely between the two systems until a predefined activity termination event is triggered, upon which the remote connection is dropped.

In a WLAN environment the W24 continues to listen on the port server listening socket, while in a dial-up environment, W24 goes offline and waits for another RING trigger.

The W24 MSEL signal (see W24 datasheet) can be lowered to GND to emulate the RING event. This is useful for testing and debugging purposes of the SerialNET connection procedure or as a means to cause W24 to activate the ring response procedure as a result of some TTL hardware signal.

Client Devices

Client devices initiate communications to a server. When a client device first sends data on its serial link, W24 (in SerialNET mode) buffers the incoming data bytes and attempts to establish a connection to a remote server. After going online, W24 performs an IP registration process according to the RRSV, RRWS, and RRMA parameters.

Once the socket connection is established, W24 transmits the buffered data collected during the connection period. The MBTB parameter dictates the maximum number of bytes to buffer. If additional bytes are received on the serial port before the connection is established, they are discarded.

W24 will dial-up the ISP to establish an Internet connection before attempting to open the server socket.

W24 closes its listening socket (if one is defined by the LPRT parameter) to avoid remote client devices from connecting during this session.

The remote server's IP and port are part of the SerialNET mode configuration parameters. Once a data connection is established, data can flow freely between the local client device and the remote server. If a connection cannot be obtained, eventually the client device's data will be discarded (similar to the case of a device transmitting serial data without a serial cable connected). Data continues to flow until a predefined activity termination event is triggered, upon which the remote connection is dropped.

Automatic SerialNET Server Wake-Up Procedure

A SerialNET client may be configured to wake up a remote SerialNET server provided it has its phone number. The SPN parameter is used to store this wakeup number.

When SPN contains a phone number and no Host Server Name and/or IP are defined, the SerialNET client tries to retrieve them from the registration e-mail of a remote SerialNET server. When characters are received from the host port, the SerialNET client dials the SerialNET server and then hangs up, causing the server to connect to its ISP, send a registration e-mail containing its IP address and local port, and open a listening socket on that port.

The client, after waking up the server, connects to its ISP and starts polling the predefined mailbox for the server's registration e-mail. Once this e-mail arrives, the client opens a socket to the IP address and port defined in the e-mail. The SWT (SerialNET Wakeup Timeout) parameter defines how long W24 will wait for this procedure to conclude before stopping. Data then flows until a predefined activity termination event is triggered, upon which the remote connection is dropped.

Transmit Packets

Data originating in the local device is buffered, packetized, and transmitted to the remote system over the network. Packets are formed as a result of meeting at least one of the following criteria:

- A predetermined number of bytes has been received from the local link (MCBF).
- The TCP/IP connection MTU was met.
- A predetermined flush character has been received (FCHR).
- A predetermined inactivity timeout event was triggered (MTTF).

Until one of these events occurs, data is buffered in the W24. When an event occurs, a packet is transmitted. The event parameters are configured by setting AT+i parameters prior to initiating SerialNET mode. When a UDP connection is used, data packets are atomic, maintaining their original size. When a TCP connection is used, packets can be combined before being actually transmitted. This follows from the stream nature of the TCP protocol. Data originating in the remote system is routed to the local device as it is made available. Flow control can be governed locally using hardware flow control only.

The PTD parameter can be used to define the number of packets to be cyclically discarded in a SerialNET mode session. When PTD>0, W24 first discards <ptd> packets before actually sending one to the SerialNET socket. This can be used to dilute repetitive information.

Completing a SerialNET Session

A SerialNET session is completed when one of the following occurs:

- The local device transmitted the disconnection string, as defined in the DSTR parameter.
- Following an inactivity timeout, as defined in the IATO parameter.

In a modem environment the W24 goes offline when the SerialNET session is terminated.

In a WLAN environment, the W24 reopens the SerialNET listening socket defined in the LPRT parameter (if it is non-zero) to service future remote client connections.

SerialNET Failed Connection

If the W24 fails to establish a SerialNET connection, SerialNET mode is deactivated for a delay period defined in the SNRD parameter.

Local Serial Port Configuration

Prior to entering SerialNET mode, W24's local serial port can be configured to comply with a wide range of devices by assigning a value to the SNSI parameter.

Serial port configuration entails settings to:

Baud rate:	300, 1200, 2400, 4800, 9600, 19200, 38400, 56K or 115K
Bits/byte:	7 or 8
Parity:	None, Even, or Odd
Stop Bit:	Must be 1
Flow Control:	None (0) or Hardware (1)

Activation Command

The W24 is forced into SerialNET mode by issuing the following command:

```
AT+i[!@]SNMD
```

If the minimal SerialNET parameters are defined, W24 replies with **I/OK** followed by **I/DONE** or **I/ONLINE** or **I/OFFLINE**.

If the W24 is online at the time this command is issued, it closes the Internet session in an orderly manner. This includes closing all open sockets and disconnecting from the ISP in a modem environment.

When W24 boots up in SerialNET mode, it sets the host serial channel to the fixed baud rate and serial interface parameters defined in the SNSI parameter. W24 in WLAN mode opens the SerialNET listening socket (if it is defined in the LPRT parameter) and, if defined, launches the web server.

In a W24 dial-up environment, the modem is polled for the RING string. If the ring-response destination e-mail parameter (RRMA) or ring-response server parameter (RRSV) contain values, W24 waits for the RING strings to subside and connects to the Internet. Once online, it sends an

e-mail to the RRMA address (if defined) and/or establishes a socket to the address in RRSV (if defined). The transmission contains the dynamic IP address received from the ISP and its listening port, on which W24 has an open listening socket, ready to serve the remote client.

W24 goes offline if one of the following events occurs:

- The remote peer closes the SerialNET socket.
- The IATO parameter is defined and times out.
- The terminating string defined in the DSTR parameter is received.

When the optional (!) (Auto-Link mode) flag is specified, W24 immediately goes online in response to the AT+i!SNMD command, opens the SerialNET listening socket (if it is defined in the LPRT parameter) or attempts to establish a socket to an HSRn address (if any HSRn is defined and LPRT is not). In this case, if one of the terminating events occurs, W24 does not go offline. Rather, the SerialNET socket is closed while W24 stays online and opens the listening or active socket again, after waiting the SNRD delay.

When the optional (@) (Deferred Connection mode) flag is specified, W24 immediately goes online in response to the AT+I@SNMD command. It opens the SerialNET listening socket (if it is defined in the LPRT parameter) but does not attempt to establish a socket to the HSRV address if it is defined. In this case, if one of the terminating events occurs, W24 does not go offline. Rather, the SerialNET socket is closed while W24 stays online and opens the listening socket again, after waiting the SNRD delay.

W24 exits SerialNET mode when one of the Escape procedures is activated.

SerialNET over TELNET

SerialNET over TELNET mode of operation opens a data socket as a TELNET socket, which allows negotiations of TELNET options over the same socket while the host is sending and receiving raw data only. This mode partially supports the RFC2217 standard.

SerialNET over TELNET mode is entered by sending the command AT+iSNMD=4 after setting W24's Host Interface to USART0 (HIF=1) or USART1 (HIF=2).

An error code - **I/ERROR (124)** - is returned upon setting the SNMD parameter to 4 while the HIF parameter is not set to either 1 or 2.

Mode of Operation

SerialNET over TELNET mode expands the Auto-Link mode (!SNMD). In this mode, W24 immediately goes online upon activating SerialNET, regardless of whether serial data has arrived or not.

If the LPRT (Listening Port) parameter is defined, W24 opens a listening port and awaits a connection, and so it acts as a TELNET server. If, on the other hand, LPRT is not defined, but HSRV (Host Server) is defined, W24 acts as a TELNET client and immediately opens a TELNET socket link to the TELNET server.

Note that, even when configured as a client, W24 still acts as a server in RFC2217. See the following section - "RFC2217 Implementation" - for a more detailed explanation.

The SerialNET over TELNET mode expands W24's TELNET client in the following aspects:

- It allows W24 to operate both as a TELNET server and client.
- It partially supports RFC2217.

In this mode, data is retrieved from the remote side as it is made available. TELNET options embedded in the server/client response stream are stripped by W24 before being turned over to the host. TELNET specifies many operational options. W24 restricts its operation mode to the minimum implementation to assure best inter-system compatibility.

Following are the TELNET options negotiated by W24. Any other options negotiated by the remote side are rejected by W24.

Option ID	Name	Value	RFC
1	echo	OFF	857
3	suppress go ahead	suppress	858
24	terminal type	VT100	1091
31	window size	whatever	1073
44	com port	partial implementation	2217

Notes:

- In SerialNET over TELNET mode, a BREAK signal that is detected on the host USART is relayed to the remote side and no reset is performed.
- If the host interface is USART1, then DSR signal changes are not detected.

RFC2217 Implementation

The RFC2217 implementation in SerialNET over TELNET mode is designed to:

- Add the ability for a remote client that connects to W24 to send COM port configuration information to the host device connected to the Internet via W24's TELNET server. The configuration changes take effect immediately, but are not preserved over software or hardware reset. The allowed configurations are the same ones available by the SNSI parameter.
- Add the ability for the host device to inform the remote side about signal changes in CTS and DSR.
- Add the ability for the remote side to change the value of the RTS and DTR signals of the host device.
- Add the ability to exchange BREAK signal indications between the host device and the remote side.

The table below lists the RFC2217 options and sub-options supported by W24. Note that W24 does not send any replies to commands or command values not supported. For more information about RFC2217, refer to the RFC2217 protocol document.

When issuing any of the following commands, W24 plays the role of a server.

Option	Allowed Values		
Baud Rate	300-115200 bps		
Data Size	7 or 8 bits		
Parity	None Odd Even		
Stop Bit	1		
Flow Control	BREAK ON BREAK OFF DTR ON DTR OFF RTS ON RTS OFF		
Notify Line State	One octet (byte). The value is a bit-level composition made up from the value table that appears in the RFC2217 protocol document. Only bit 4 is supported, value 16, meaning BREAK-detect error.		
Notify Mode State	One octet (byte). The value is a bit-level composition made up from the value table that appears in the RFC2217 protocol document. Only the following bits are supported:		
	Bit Position	Value	Meaning
	5	32	Data-Set-Ready Signal State
	4	16	Clear-To-Send Signal State
	1	2	Delta Data-Set-Ready
	0	1	Delta Clear-To-Send

File Transfer Protocol (FTP) Theory of Operation

Introduction

The FTP client component in W24 extends W24's general-purpose sockets to incorporate an additional, dedicated socket for FTP activities. From the host's perspective, the FTP capabilities are a logical extension of the capabilities of e-mail and direct socket manipulation.

As in all other W24 protocol implementations, host involvement in the specifics of FTP is minimal. W24 needs to deal with non-standard FTP issues, such as possible differences between FTP server responses, on its own. Multi-stage FTP protocol sequences are atomized under W24 control to minimize complexity and need for host processor intervention.

The FTP protocol is described in RFC 959.

W24 Family FTP Client Command Set

- 
- Open FTP link to FTP Server
 - Retrieve File List from Server
 - Change Directory on Server
 - Retrieve File Contents from Server
 - Open a New File on Server
 - Open an existing File on Server for Append
 - Send Binary Data to an open File on Server
 - Close a File on Server After Binary Data Send
 - Delete File on Server
 - Close FTP Session

W24 FTP Client Operation Mode

FTP specifies several operational modes. The RFC calls for a minimum implementation, which should be observed by all FTP servers. W24 restricts its operation mode to the minimum implementation to assure best intersystem compatibility.

Character Types:	ASCII Non-print
Structure:	File
Mode:	Stream

FTP Command Socket

The FTP command socket is normally on port 21 (decimal) of an FTP server. However, other ports can be specified to support special cases.

FTP Receive Flow

Figure 2-8 shows the FTP receive flow.

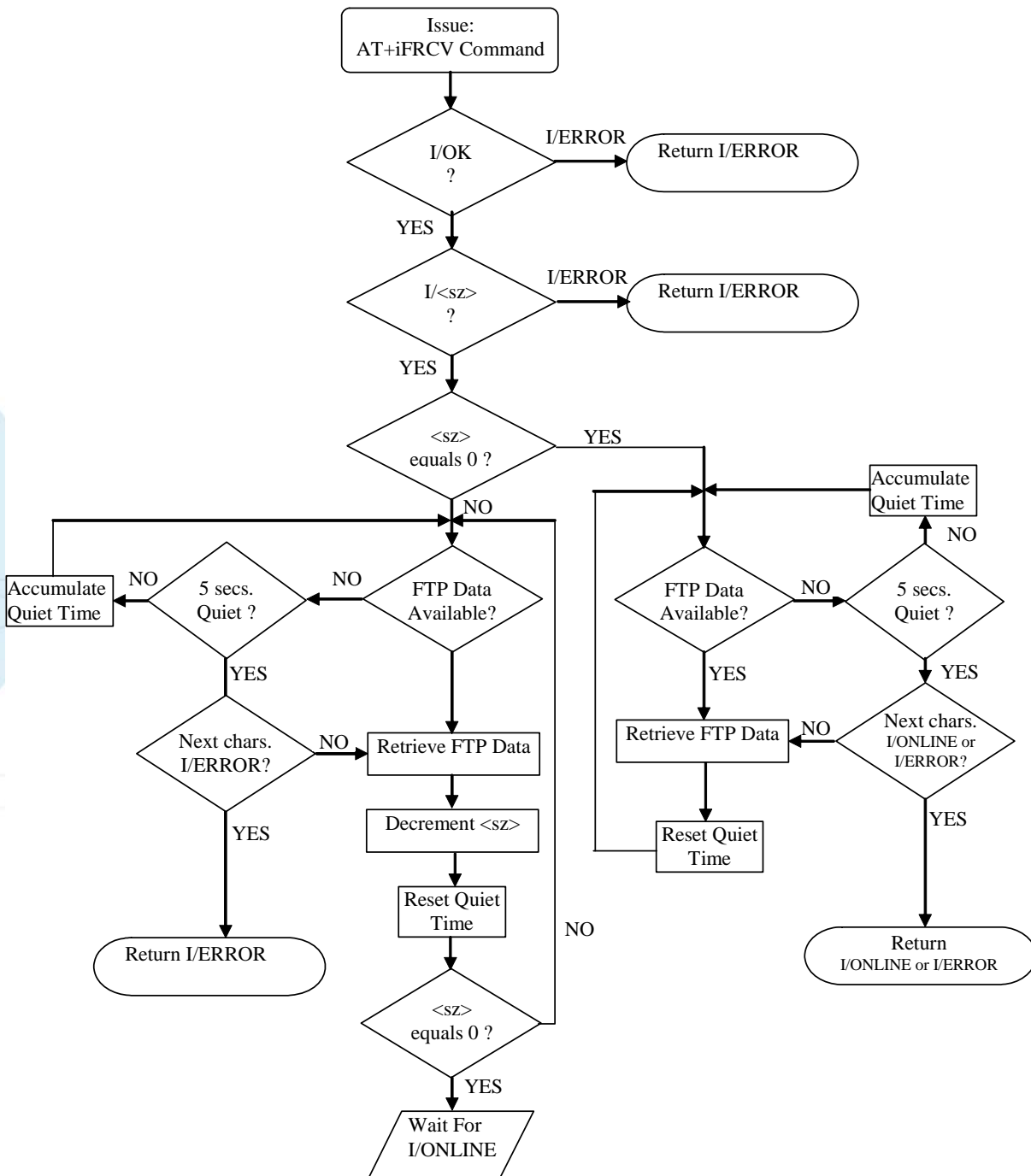


Figure 2-8: FTP Receive Flowchart

Telnet Client Operation

There are four operation modes for most Telnet applications, namely, half-duplex, character at a time, line at a time, and line mode.

W24 incorporates two methods to send data to the remote Telnet server: One line at a time, namely, an AT+i command (+iTSND) is used to send a single (CR/LF terminated line to the Telnet server); and Binary Transmission, where an AT+i command (+iTBSN) is used to send an arbitrary amount of binary data.

Data is retrieved from the remote Telnet server as it is made available. Embedded Telnet options in the server's response stream are stripped by W24 before being turned over to the host.

Telnet specifies many operational options. W24 restricts its operation mode to the minimum implementation to assure best intersystem compatibility.

Following are the Telnet options negotiated by W24:

Option ID	Name	Value	RFC
1	echo	OFF	857
3	suppress go ahead	suppress	858
24	terminal type	VT100	1091
31	window size	whatever	1073

Any other options negotiated by the Telnet server are rejected by W24.

Secure Socket Protocol Theory of Operation

Introduction

W24 implements an SSL3/TLS1 client socket connection. When connecting to an SSL3/TLS1 server, W24 negotiates an SSL3/TLS1 secure connection. During the negotiation process, the server identifies itself to the client (W24) by sending a certificate. The certificate's main purpose is to allow W24 to determine that the server is indeed the server it claims to be.

To fulfill its purpose, the certificate contains the server's ID information (name, address, description, etc.) and its public key. It also contains a digital signature, signed by a third-party called a Certificate Authority (CA), which authenticates this information. The client must trust the CA in order to accept its signature on a certificate. Furthermore, the trust relationship between the client and the CA must be established prior to the communication session and preferably using alternate methods. W24's CA parameter is used to store the CA's certificate. Once a trusted CA's certificate is stored on W24, it will accept certificates signed by that CA from SSL3/TLS1 servers it connects to.

Generating Certificates for Use with Servers

The most common way to obtain a certificate is to buy one from a commercial certificate authority. This results in a public key that has been digitally signed by a trusted third-party. Any clients receiving this certificate can be sure they are communicating with an authentic entity. However, in a trusted environment, it is possible to create an in-house CA and to self-sign the certificate.

Commercial CA's are usually preferred when connecting to multiple unknown servers. However, in distributed system configurations where not more than a handful of secure servers are deployed; an in-house CA is probably more appropriate and just as secure.

Several free software packages are available for generating certificates. The following sections describe how to use the standard OpenSSL package to generate certificates. They contain instructions on how to obtain your own certificates suitable for use with servers to which W24 will connect. Furthermore, most FTP servers that support SSL3 include a certificate generation utility that may be used to generate self-signed certificates. The self-signed certificate is part of the FTP server's configuration and may also be loaded into W24 to allow it to connect to that FTP server using SSL3 secure sockets.

Using the OpenSSL Package to Create Certificates

OpenSSL is a widely used SSL toolkit available for free download at <http://www.openssl.org>. The SSL toolkit contains source code that can be compiled for Unix, Linux, or Windows. Pre-compiled binaries are also available for these platforms. OpenSSL comes with a command line utility for generating keys, creating CA's, and creating certificates.

The following instructions assume the OpenSSL package has been installed and configured properly on your machine. The instructions walk you through using OpenSSL to create an in-house Certificate Authority, sign your own certificates, and generate the proper requests in order to receive a signed certificate from a commercial CA. The signed certificates can then be installed on servers to which W24 will connect in a secure (SSL3/TLS1) manner.

Creating a Certificate Authority

The certificate generated using the following steps can be used in deployed systems, in which you are the trusted authority. Users of these certificates can be confident of your identity. For example, W24 devices communicating with servers that are setup and configured by the device vendor can secure their communications using certificates signed by the vendor-created Certificate Authority.

In order to store the files to be generated, create a new directory named *testCA*.

Open a command will (on Windows, enter *cmd* in the Start > Run dialog box), change the command shell's working directory to *testCA* and follow these instructions:

Creating the CA Environment

The creation of a CA produces several files that must be preserved throughout the lifecycle of the CA. You can sign an unlimited number of certificates using a single CA. These files are written to each time you sign a certificate.

1. Under the *testCA* directory create sub-directories *certs* and *private*.
2. Create a new file named *serial*. In this file enter the numerals '01' and save the file.
3. Create an empty file named *index.txt*.

Creating the Test CA Configuration File

Whereas you can enter all configuration information in a command line, creating a configuration file makes these steps easier to reproduce and allows you to save the options used to create a CA.

1. Create a new file named *CAnf.ca* using a text editor of your choice.
2. Add the following basic CA configuration information:

```
[ ca ]
default_ca = CA_default

[ CA_default ]
dir = /testCA
certificate = $dir/cacert.pem
database = $dir/index.txt
new_certs_dir = $dir/certs
private_key = $dir/private/caprivkey.pem
serial = $dir/serial
default_crl_days = 7
default_days = 365
default_md = md5
policy = CA_default_policy
```

```
x509_extensions = certificate_extensions
```

```
[ CA_default_policy ]
commonName = supplied
stateOrProvinceName = supplied
countryName = supplied
emailAddress = supplied
organizationName = supplied
organizationalUnitName = optional
```

```
[ certificate_extensions ]
basicConstraints = CA:false
```

```
[ req ]
dir = /testCA
default_bits = 1024
default_keyfile = $dir/private/caprivkey.pem
default_md = md5
prompt = no
distinguished_name = root_ca_DN
x509_extensions = root_ca_extensions
```

```
[ root_ca_DN ]
commonName = Common Name                # Server name or YOUR
name
stateOrProvinceName = My State
countryName = US                        # 2 Letter Code
emailAddress = myemail@mydomain.com     # Your Email Address
organizationName = My Organization
organizationalUnitName = Organization Unit # Unit Name (ie,
section)
```

```
[ root_ca_extensions ]
basicConstraints = CA:true
```

Note that both `dir` entries under `[CA_default]` and `[req]` must be set to the path to the `testCA` directory created earlier. The `root_ca_DN` section can be changed to enter information specific to your organization.

Creating a Self-Signed Root Certificate

A certificate authority is essentially a self-signed root certificate. This root certificate is used to respond to new certificate requests to create a signed certificate. In this case, W24 is both the CA and the originator of the certificate request, so no identity verification issues exist. In a more typical situation, however, a CA can only be trusted if it performs sufficient background checks into the originator of the certificate request to verify its identity.

1. Set the OPENSSL_CONF system environment variable to point to the newly created configuration file.
 - On Linux\Unix, type the following:

```
OPENSSL_CONF=/testCA/CAcnf.ca
export OPENSSL_CONF
```
 - On Windows, type the following:

```
set OPENSSL_CONF=C:\testCA\CAcnf.ca
```
2. Enter the command for generating the self-signed root certificate (all text is a single command typed on one line):

```
openssl req -x509 -newkey rsa:1024 -out cacert.pem -outform PEM
```
3. You are prompted to enter a PEM pass phrase. This is your password to the CA private key. It is essential for the security of the system that both this password and the CA private key are kept secret.

An encrypted *caprivkey.pem* file, which is the private key for the CA is now stored under the *private* sub-directory. The self-signed *cacert.pem* file is stored under the top-level *testCA* directory.

The *cacert.pem* certificate can be used to sign new certificate requests as detailed in the following steps. Alternatively, the *cacert.pem* certificate can be used as-is in a server system if the single level hierarchy is considered sufficient.

The *cacert.pem* certificate has to be loaded into W24's CA parameter to enable W24 to trust and communicate securely with servers whose certificate is *cacert.pem* or that use certificates **signed** with *cacert.pem*.

Signing a Certificate with a CA Certificate

Creating a Certificate Request

Now that the CA has been created, you can use it to sign new certificates. In this example, W24 plays the role of the CA, the certificate subject, and the end-user of the certificate, so no trust issues exist. A typical process, however, involves communication between the certificate subject (you) and a trusted CA. Usually someone wishing to issue certificates to end-users would generate a certificate request file and submit it to the administrators of a CA. Once the administrators of the CA have determined the request to be valid, a self-signed root certificate would be used to sign the certificate request and create a new certificate to be returned to the originator of the request, and eventually to the end-user.

1. Reset the OPENSSL_CONF environment variable to the default *openssl.cnf* file. Generating a request has nothing to do with a CA before it is actually submitted. It is safe to point OPENSSL_CONF to the default configuration file because it will force the request

command to prompt the user for all information regarding the certificate request. Set the environment variable to the default file by typing the following:

- On Linux\Unix:


```
OPENSSL_CONF=/OpenSSL/apps/openssl.cnf
export OPENSSL_CONF
```
 - On Windows:


```
set OPENSSL_CONF=C:\OpenSSL\bin\openssl.cnf
```
2. Generate the request with the following single line command and answer all questions at the prompt:


```
openssl req -newkey rsa:1024 -keyout myprivkey.pem -keyform
      PEM -out myreq.pem -outform PEM
```

If you do not want an encrypted private key, add *-nodes* to the above command. At the conclusion of this step two new files are created. The *myprivkey.pem* file contains the encrypted private key. This file must never be shared, not even with the CA. The other file is the certificate request file, *myreq.pem*, which will be used by the CA to create the final signed certificate.

Using the Test CA to Issue the Certificate

The final step of the process is to use the CA self-signed certificate to sign the certificate and return it to the originator of the request (subject).

1. Reset the OPENSSL_CONF system environment variable to reference the CA configuration file again.
 - On Linux\Unix type the following:


```
OPENSSL_CONF=/testCA/CAcnf.cnf
export OPENSSL_CONF
```
 - On Windows type the following:


```
set OPENSSL_CONF=C:\testCA\CAcnf.cnf
```

Make sure that the request file is in the current directory and run the following command. The PEM password you are prompted to enter is the password for the CA private key file:

```
openssl ca -in myreq.pem
```

You will be requested to enter the pass phrase for the CA private key that was generated above. Enter the pass phrase to continue.

Answer 'y' at the next two prompts, then at the conclusion of this step several files are updated and a new certificate is created.

The new certificate can be found in the *certs* sub-directory. It is named as the serial number it is associated with by the CA. The file can be renamed, but the *.pem* extension must be preserved for clarity. The *serial* file itself increments its count for the next certificate request and the *index.txt* file shows a record of the creation. The new certificate file and the *myprivkey.pem* file are now suitable for use by an SSL server to which W24 needs to connect. As mentioned above, the W24 +iCA parameter must contain the CA certificate *cacert.pem* used to sign the server's certificate.

Remote AT+i Service

Introduction

The LATI parameter allows configuring W24 to maintain a communication channel that supports interacting with W24 from a remote location using the AT+i command set as if the commands are administered through the local serial port. When LATI is set to a non-zero value, W24 opens a TCP listening socket on port *<LATI>*. In a dial-up environment, this occurs only after the PPP connection is established. This listening socket can be used to connect to W24's remote AT+i service.

Remote AT+i Commands

When a remote client connects to W24's LATI socket, W24 redirects the socket's data flow to the AT+i parser, in effect allowing the socket to take over the parser. Any data coming from the socket is processed by W24 as if it came from the host serial port and the replies are returned to the socket instead of being sent to the host serial port. W24 replies with an I/BUSY to commands coming from the host serial port, while the remote client is connected.

An exception to this is the (+++) escape sequence. On detection of (+++) from the host serial port, W24 closes the remote connection and reboots.

If W24 was in the process of performing some Internet activity initiated by the host at the time the remote client connected, W24 allows this activity to end and the final reply to reach the host before passing control over the parser to the remote client.

Closing a Remote AT+i Session

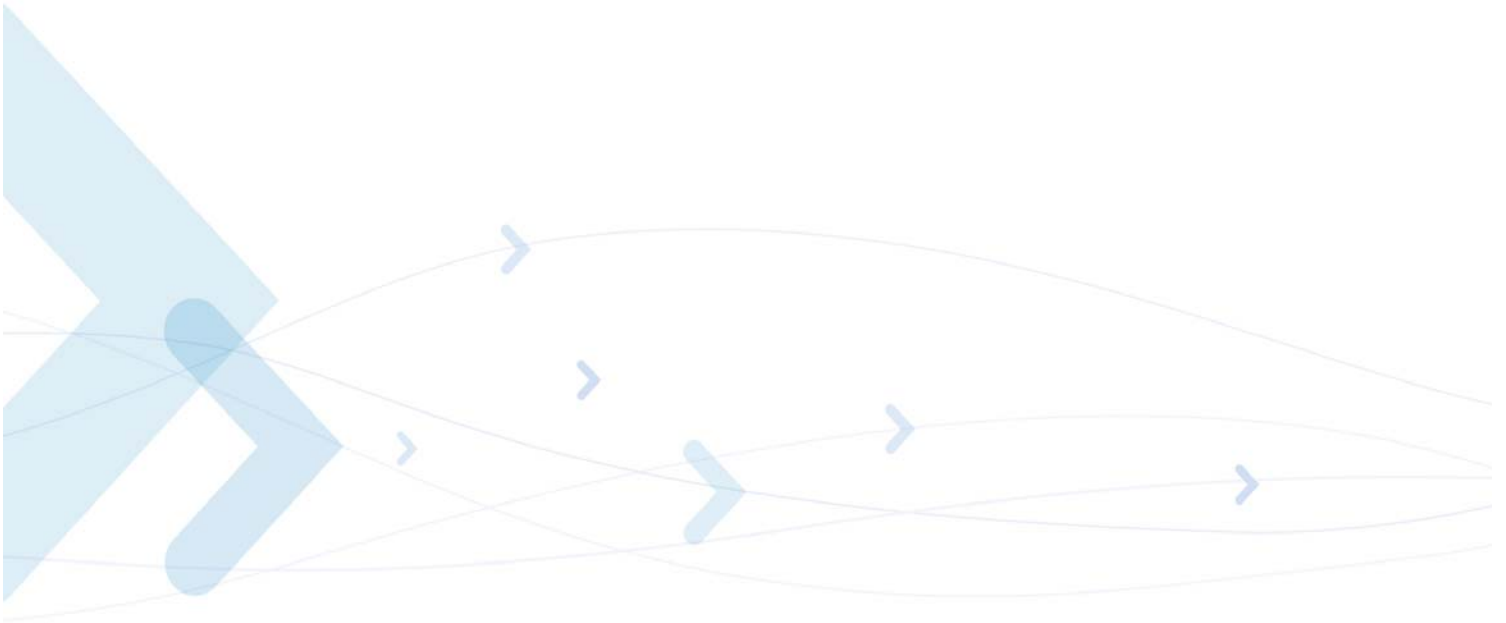
To close a remote AT+i session, the remote client can choose to issue AT+iDOWN via the socket. In response to this, W24 restarts. Only I/OK is returned over the socket before it is closed by W24. Alternatively, the remote client can close the socket in order to disconnect, leaving W24's Internet session as-is. In the latter case, W24 returns control over the parser to the local host port. The LATI listen remains active, available to service additional remote connections. After a LATI session is closed, the LSR (last session error) web parameter contains the value 096 to indicate that a LATI session has been disconnected.

Note: (+++) sent over the LATI socket is not recognized as an escape sequence.

Caveats and Restrictions

- When W24 in dial-up mode is in auto baud rate detection mode (after re-starting with BDRF=a or in response to the AT+iBDRA command), a remote AT+i session cannot be established, even if the LATI parameter contains a port value.
- In W24 WLAN the remote AT+i service is available, even if W24 WLAN is in auto baud rate detect mode. However, once the remote AT+i connection is established, W24 WLAN will no longer be in auto baud rate mode and the host will be able to send the (+++) escape sequence only at 9600 baud, if it needs to close the remote session. W24 WLAN will then return to auto baud rate detect mode when and if the local host or the remote client close the LATI session, in effect re-starting W24 WLAN.

- During a remote AT+i session, the remote client taking over the parser cannot make use of W24's mechanisms of Hardware or Software flow control, which exist for the local host port. The only mechanism W24 will use in this mode is TCP level flow control (using the TCP window).
- In W24 WLAN, AT+IBDR or AT+IBDRA will return I/OK but will not initiate a baud rate detection process.
- The remote AT+i commands socket cannot be used to send AT+i command to W24 when W24 is in SerialNET mode.



Nonvolatile Parameter Database

Parameter Descriptions

See [Table 2-6](#).

Table 2-6: Nonvolatile Parameter Database

Parameter	Type	Range	Default	Description
Operational				
XRC	Byte	0..4	4	Extended Return Code. Same as ATXn.
DMD	Byte	0..2	0	Modem Dial Mode: ATD< <i>m</i> > <i>m</i> : Tone (0); Pulse (1); None (2)
MIS	String	126 chars	"AT&FE0 V1X4Q0& D2M1L3\r"	Modem initialization string. May contain several consecutive AT commands.
MTYP	Byte	0..11	0	Modem Type Designator.
WTC	Byte	0..255	45	Wait Time Constant. Initialization constant for modem's S7 register. Defines a timeout constant for a variety of modem activities.
TTO	INT	0..3600	0	TCP Timeout. Number of seconds to wait before returning a timeout error on a TCP transaction.
PGT	Unsigned INT	0-65535	0 [mSec]	Timeout to resend a PING request.
MPS	Byte	0..3	0 (1500)	Max PPP Packet Size.
TTR	INT	1000..65535	3000 [mSec]	Timeout to resend an unacknowledged TCP packet over PPP, in milliseconds.
BDRF	Byte	3..9 'a' 'h'	'a' (Auto)	Sets the W24<->Host to a fixed baud rate.
BDRM	Byte	3..9 'a' 'h'	'a' (Auto)	Sets the W24<->modem baud rate to a fixed baud.
AWS	Byte	0..3	0	Sets flag to define web server activation. 0 (web server disabled), 1 2 3(web server enabled).
LATI	INT	0-65535	0 (Disabled)	Remote AT+i Service, port number.
FLW	Byte	0..7	0 (S/W)	Flow Control Mode.
CPF	Byte	0..1	1 (WLAN)	Sets Communication Platform: Modem (0); WLAN (1).
PSE	Byte	0..255	0 (Disabled)	Sets Power Save Mode: Disabled(0); idle time in seconds before activating Power Save mode (1..255).
SDM	Byte	0..7	0 (All Enabled)	Service Disable Bitmap.

Table 2-6: Nonvolatile Parameter Database (Cont.)

Parameter	Type	Range	Default	Description
DF	Byte	0..1	0	IP Protocol Don't Fragment Bit.
CKSM	Byte	0..1	0 (Disabled)	Sets checksum mode.
HIF	Byte	0..5	0	Sets host-to-W24 interface.
MIF	Byte	1..5	2 (USART1)	Sets W24-to-modem interface.
ADCL	Byte	0-255	0	A/D Converter base level.
ADCD	Byte	0-255	0	A/D Converter delta.
ADCT	INT	0-65535	0	Time interval between queries of the A/D Converter's register.
ADCP	INT	0-96	0	W24's I/O pin to be asserted by the A/D Converter's polling mechanism.
RRA	Byte	0-6	0	W24 readiness indication.
RRHW	INT	0-96	0	W24 readiness HW pin
ISP Connection				
ISPn	Phone #	96 chars	NULL	ISP's access phone number. <n>: 1..2
ATH	Byte	0..2	1 (PAP)	Use CHAP (2), PAP (1) or Script (0) authentication.
USRN	String	64 chars	NULL	ISP Connection User Name.
PWD	String	64 chars	NULL	ISP Connection Password.
RDL	Byte	0..20	5	Number of Redial tries.
RTO	Byte	0..3600	180	Timeout before redialing [seconds].
Server Profiles				
LVS	Byte	0..1	1 (YES)	Leave on Server: 1(YES), 0 (NO)
DNSn[p]	IP address		0.0.0.0	Domain Name Server IP address <n>:1..2
SMTP[p]	String	64 chars	NULL	SMTP Server Name.
SMA	Byte	0..1	0 (None)	Define SMTP Authenticated Method: 0 (None) 1(Login authentication)
SMU	String	64 chars	NULL	SMTP Authentication User Name.
SMP	String	64 chars	NULL	SMTP Authentication Password.
POP3[p]	String	64 chars	NULL	POP3 Server Name.
MBX	String	64 chars	NULL	Mailbox User Name.
MPWD	String	64 chars	NULL	Mailbox Password.
NTSn	String	64 chars	NULL	Network Time Server name <n>: 1..2
NTOD	Byte	0..1	0 (Disabled)	Network time-of-day retrieval flag.
GMTO	Byte	-12..12	0	W24 location's GMT Offset.

Table 2-6: Nonvolatile Parameter Database (Cont.)

Parameter	Type	Range	Default	Description
DSTD	String	64 chars	NULL	Sets W24's Daylight Savings transition rule.
PDSn	String	64 chars	NULL	Sets W24's PING Destination servers, used for online status verification.
PFR	INT	0-65535	0 (Disabled)	Sets PING destination server polling frequency.
User Fields				
UFn	String	128 chars	NULL	User Storage field and Macro Substitution <n>: 01..12
E-Mail Format				
XFH	Byte	0..1	1	Transfers e-mail headers. 1 (Enable) 0 (Disable).
HDL	Byte	0..255	0 (no limit)	Limits number of header lines retrieved.
FLS	String	64 chars	NULL (no filter)	Filter string must exist in message header to Qualify for Retrieve.
DELF	String	64 chars	None	E-mail Delete Filter.
SBJ	String	96 chars	NULL	Contents of the e-mail subject field.
TOA[n]	String	64 chars	NULL	E-mail Addressee.
TO	String	96 chars	NULL	Addressee Description/Name in e-mail header.
REA	String	64 chars	NULL	Returns e-mail address.
FRM	String	96 chars	NULL	Sender Description/Name in e-mail header.
CCn	String	64 chars	NULL	Alternate Addressee (CC: field) <n>: 1..4
BDY	Text lines	96 chars	NULL	Textual body contents for MIME-encapsulated e-mail messages.
MT	Byte	0..4	4 (app.)	Media Type: 0: Text; 1: Image; 2: Audio; 3: Video; 4: application
MST	String	64 chars	octet-stream	Media Subtype String. For a list see “MIME Content Types and Subtypes” on page A-1.
FN	String	64 chars	None	Attachment File Name (inc. extension). If a file name is not defined, W24 generates a unique filename.
IP Registration				
RRMA	String	64 chars	NULL	Sets the e-mail address to use for dynamic IP address registration after going online.
RRSV	String	64 chars	NULL	Sets the server name/IP and port to contact for dynamic IP address registration after going online.
RRWS	String	128 chars	NULL	Sets the web server URL used for dynamic registration after going online.

Table 2-6: Nonvolatile Parameter Database (Cont.)

Parameter	Type	Range	Default	Description
RRRL	String	64 chars	NULL	Sets the Return Link IP address to use when performing an IP address registration behind a NAT.
HSTN	String	64 chars	NULL	W24's Network Host Name, included in all IP registration methods. W24 WLAN will be registered in DNS through DHCP Server.
HTTP				
URL	String	128 chars	None	URL string used for subsequent +iRLNK and +iSLNK commands.
CTT	String	64 chars	NULL	Defines the "Content-type" field sent in the POST request by the +iSLNK command.
WPWD	String	64 chars	NULL	Password for restricting host parameter updates via a web browser.
RAS Server				
RAR	Byte	2..20	4	Number of RINGs after which W24 will activate its internal RAS Server.
RAU	String	64 chars	NULL	RAS Login User Name.
RAP	String	64 chars	NULL	RAS Login Password.
LAN				
MACA	String	12 chars	MAC address	MAC address assigned to W24.
DIP	Default IP address		0.0.0.0	Default IP address stored in W24's nonvolatile memory.
IPA	IP address		0.0.0.0	IP address assigned to W24.
IPG	IP address		0.0.0.0	IP gateway address assigned to W24.
SNET	IP address		0.0.0.0	Subnet address assigned to W24.
802.11 b/g Wireless LAN				
WLCH	Byte	0..13	0	Wireless LAN Communication Channel in ad-hoc mode.
WLSI	String	32 chars	NULL	Wireless LAN System Set ID.
WLWM	Byte	0..2	0 (Disabled)	Wireless LAN WEP Mode.
WLKI	Byte	1..4	1	Wireless LAN Transmission WEP Key Index.
WLKn	Hex Key String	26 chars	NULL	Wireless LAN WEP Key Array.
WLPS	Byte	0..5	0	Marvell WiFi chipset Power Save mode dose time.
WLPP	String	8-63 chars	NULL	Wireless LAN WPA- PSK pass phrase.
WSEC	Byte	0..1	0 (WPA security)	Wireless LAN WPA security option.

Table 2-6: Nonvolatile Parameter Database (Cont.)

Parameter	Type	Range	Default	Description
WROM	Byte	0..1	0	Enable Roaming mode.
WPSI	INT	1-3600	5	Periodic scan for APs interval.
WSRL	Byte	0-255	10	Roaming mode SNR low threshold.
WSRH	Byte	0-255	30	Roaming mode SNR high threshold.
WSIn	String	32 chars	" (Empty)	WLAN SSID for multiple SSIDs.
WPPn	String	8-63 chars	" (Empty)	Pre-shared key passphrase for multiple SSIDs.
WKYn	String	26 chars	" (Empty)	WLAN WEP key for multiple SSIDs.
WSTn	Byte	0..4	0	WLAN security type for multiple SSIDs.
SerialNET Mode				
HSRV or HSRn	String	64 chars	NULL	Set the remote host server name/IP and port.
HSS	String	3 chars	NULL	Switches among three possible HSRV parameters.
DSTR	String	8 chars	NULL	Set the disconnection string template.
LPRT	Unsigned INT	0-65535	0	Set the SerialNET mode listen socket.
MBTB	INT	0-2048	0	Max bytes to buffer while W24 is establishing a connection.
MTTF	Unsigned INT	0-65535	0 (None)	Max inactivity timeout in milliseconds before flushing the SerialNET socket.
FCHR	Byte	1 char	0 (None)	Flush character. When received, SerialNET socket will be flushed.
MCBF	INT	0-1460	0 (None)	Max. characters before flushing the SerialNET socket.
IATO	INT	0-32768	0 (None)	Inactivity timeout in seconds before closing the SerialNET connection.
SNSI	String	9 chars	"5,8,N,1,0"	SerialNET mode Serial interface configuration. Defines baud, bits, parity, stop and flow control.
STYP	Byte	0..1	0 (TCP)	Set the SerialNET mode socket type. 0 (TCP) or 1 (UDP).
SNRD	INT	0..3600	0 (No Delay)	Delay time in seconds before re-enabling SerialNET mode after a failed connection.
SPN	String	96 chars	NULL	SerialNET Phone Number to wake-up SerialNET Server.
SDT	Byte	0..255	20	SerialNET Dial Timeout. When waking up a SerialNET server, W24 will hangup after SDT seconds have elapsed.
SWT	INT	0..65535	600	SerialNET Wake-up Timeout. Number of seconds to allow for the SerialNET server wake-up procedure.

Table 2-6: Nonvolatile Parameter Database (Cont.)

Parameter	Type	Range	Default	Description
PTD	INT	0..65535	0 (No Filter)	Specifies the number of Packets to Drop during a SerialNET session.
Remote Firmware Update				
UEN	Byte	0..1	0	Remote Firmware Update flag.
Remote Parameter Update				
RPG	String	64 chars	NULL	Remote Parameter Update Group/Password.
Secure Socket Protocol (SSL3/TLS1)				
CS	Byte	0, 4, 5, 10, 47, 53	0 (propose all)	Set the cipher suite to be used during SSL3/TLS negotiations.
CA	String	1300 chars	NULL	Set W24's SSL3/TLS trusted Certificate Authority (CA).
CERT	String	4 Kbyte	NULL	Set W24's SSL3/TLS certificate.
PKEY	String	4 Kbyte	NULL	Set W24's private key.
DHCP Server				
DPSZ	Byte	0-255	0 (DHCP server off)	Set number of addresses in W24's IP pool.
DSLT	INT	0-65535	0 (No limit)	Define lease time, in minutes, granted when assigning IP addresses to clients.
iRouter Mode				
ARS	Byte	0..1	0	Causes W24 to automatically enter iRouter mode upon power-up or soft reset.

+iFD - Restore All Parameters to Factory Defaults

Syntax: AT+iFD

Restore W24's non-volatile parameter database values to factory defaults.

Each of W24's nonvolatile parameters, described in the following section, has an associated default value. This command restores all parameters to their factory default values.

This command disables W24's DHCP client. In order to re-activate the DHCP client process, you need to perform a HW or SW reset.

This command also resets W24's active IP address stored in the IPA parameter.

An exception to the above are the MIS (Modem Init String), RPG (Remote Parameter Group/Password) and CPF (Communications Platform) parameters, which will always retain the last set value.

Result Code:
I/OK After restoring parameters to factory default values.

Operational Parameters

+iXRC - Extended Result Code

Syntax: AT+iXRC=*n*
Extended Result Code. Same as ATX*n*. This command selects which subset of the result messages will be used by the modem to inform the Host of the results of commands.

Parameters: *n*=0..4

Command Options: For a detailed description of the command options see the ATX*n* command in the AT command set manual for the modem in use.

Default: 4

Result Code:
I/OK If *n* is within limits.
I/ERROR Otherwise.

AT+iXRC~*n* Temporarily sets the Extended Result Code for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.

AT+iXRC? Report the current Extended Result Code used when dialing the ISP. The reply is followed by **I/OK**.

AT+iXRC=? Returns the message "0-4" followed by **I/OK**.

+iDMD - Modem Dial Mode

Syntax: AT+iDMD=*n*
Permanently sets the modem dial mode to Tone, Pulse or none. This parameter defines the dial character *m* used when issuing the ATD*m* dial command to the modem.

Parameters: *n*=0..2

Command Options:

n=0 Use Tone dialing (*m*=T).
n=1 Use Pulse dialing (*m*=P).
n=2 Use modem's default dialing (*m*=").

Default:	0 (Tone Dialing).
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iDMD~ <i>n</i>	Temporarily sets the modem dial mode for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iDMD?	Reports the current modem dial mode used when dialing the ISP. The reply is followed by I/OK .
AT+iDMD=?	Returns the message "0-2". The reply is followed by I/OK .

+iMIS - Modem Initialization String

Syntax:	AT+iMIS= <i>str</i> [; <i>str</i> ...]
	Sets the Modem Initialization String.
Parameters:	<i>str</i> =Modem initialization string
Command Options:	
<i>str</i> =	Empty: No modem initialization string defined.
<i>str</i> < <i>string</i> >	<i>string</i> will be used as the modem initialization string. If <i>string</i> contains special characters, such as quotation marks (' or "), these may be included in <i>string</i> by prefixing each special character with a backslash (\). For example: "AT+CGDCONT,\"IP\",,\"INTERNET\"". <i>string</i> must include the AT prefix and the modem reply is expected to include 'OK'. MIS may include several consecutive modem commands separated by a semicolon. Each command must begin with 'AT' and its modem reply must include 'OK'. W24 will send each 'AT' command separately, followed by <CR> and wait for the 'OK' before proceeding.
Default:	'AT&FE0V1X4Q0&D2MIL3' <i>Note:</i> This default value is shipped from the factory.
Result Code:	
I/OK	If <i>str</i> is an empty or a legal string.
I/ERROR	Otherwise.
AT+iMIS~ <i>str</i> [; <i>str</i> ...]	Temporarily sets the modem initialization string to <i>str</i> [; <i>str</i> ...]. The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iMIS?	Reports the current modem initialization string. If the modem initialization string is empty, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iMIS=?	Returns the message 'String' followed by I/OK .

+iMTYP - Set Type of Modem Connected to W24

Syntax: AT+iMTYP=*n*

Sets the modem type.

Parameters: *n*=0..9

Command Options:

<i>n</i> =0	Standard, Hayes compatible, dialup modem.
<i>n</i> =1	Silicon Laboratories Si2400 modem. See note below.
<i>n</i> =2	Standard GSM modem.
<i>n</i> =3	AMPS CM900 modem.
<i>n</i> =4	Falcom GSM modem.
<i>n</i> =5	Silicon Laboratories high-speed modems Si2414/33/56.
<i>n</i> =6	Standard 2400 baud modem (increased timeout).
<i>n</i> =7	GSM 536 modem (packets limited to 536 bytes).
<i>n</i> =8	CDPD cellular modem.
<i>n</i> =9	Wavecom Fastrack cellular modem.
<i>n</i> =10	SiLABs World modem.
<i>n</i> =11	Telit GE862-PY cellular modem.
+100	Add 100 to any modem type to prohibit W24 from issuing an ATZ to the modem before dialing the ISP when an Internet session is activated. This is useful if the modem needs to be initialized manually before an Internet session. Note that an ATZ will be issued when the session is terminated.

Default: *n*=0 Standard, Hayes compatible, dialup modem

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iMTYP? Returns current modem type designator followed by **I/OK**.

AT+iMTYP=? Returns the message "0-11" followed by **I/OK**.

Note 1:Configuring the W24 to work with Silicon Laboratories Si2400:

- 1 AT+iMTYP=1
- 2 AT+iMIS=""
- 3 AT+iBDRF=3
- 4 AT+iBDRM=3

Note 2:Configuring the W24 to work with GPRS modems:

- 1 AT+iMTYP=2 - GSM/GPRS modem type
- 2 AT+iXRC=0 - blind dialing

3 AT+iISP1=<ISP/Provider dial number> (usually *99**1#)

4 AT+iMIS="AT+CGDCONT=1,IP,<Proxy>"

Note 3: Changing from modem type 4 (Falcon GSM):

When W24 is configured with MTYP=4, the MTYP parameter must first be changed to the special value 99 before it can be changed to some other value.

Note 4: Working with SiLABS World modems:

With modem type 10 selected, W24 waits 300msec after issuing ATZ at the end of a session before issuing additional commands to the modem.

+iWTC - Wait Time Constant

Syntax: AT+iWTC=*n*

This parameter is used to set the modem register S7 to the required value (using the "ATS7=*n*" command).

Parameters: *n*=0..255

Command Options: The WTC parameter defines a timeout constant for a variety of modem activities. For a detailed description of this parameter, see the ATS7=*n* command in the AT command set manual for the modem in use.

Default: 45 seconds

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iWTC~*n* Temporarily sets the Wait Time Constant value for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.

AT+iWTC? Reports the current Wait Time Constant used. The reply is followed by **I/OK**.

AT+iWTC=? Returns the message "0-255". The reply is followed by **I/OK**.

+iTTO - TCP Timeout

Syntax: AT+iTTO=*n*

Sets the number of seconds W24 allots an Internet transaction to complete before returning the timeout error.

Parameters: *n*=0..3600 seconds

Command Options: The TTO parameter defines the timeout constant for Internet transactions. W24 will return with a timeout error for any TCP/UDP/IP transaction that didn't complete properly within $n \pm 10\%$. Timeout measurement is defined between receipt of an AT+i command and a W24 response to the host. In dial-up environments, timeout measurement begins only after establishing a PPP connection. Furthermore, an additional 10-15 seconds may be required to allow the W24 to disconnect the modem. $n=0$ is a special case where internal timeout constants will be used.

Default: 0 (use W24's factory default timeout values)

Result Code:

I/OK If n is within limits.

I/ERROR Otherwise.

AT+iTTO~ n Temporarily sets the Internet transaction timeout value for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.

AT+iTTO? Reports the current Internet transaction timeout used. The reply is followed by **I/OK**.

AT+iTTO=? Returns the message "0-3600" followed by **I/OK**.

+iPGT - PING Timeout

Syntax: AT+iPGT= n

Sets the timeout in milliseconds, after which W24 will reissue a PING request that has not been replied to.

Parameters: $n=0..65535$ milliseconds

Command Options: After issuing a PING request, in response to the AT+iPING command, W24 will wait up to n milliseconds for a reply. If a reply is not received, W24 will reissue the PING request. $n=0$ is a special case where a timeout of 2 seconds is used.

Default: 0 (use W24's factory default 2 seconds timeout)

Result Code:

I/OK If n is within limits.

I/ERROR Otherwise.

AT+iPGT~ n Temporarily sets the PING transaction timeout value for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.

AT+iPGT? Reports the current PING transaction timeout used. The reply is followed by **I/OK**.

AT+iPGT=? Returns the message "0-65535" is followed by **I/OK**.

+iMPS - Max PPP Packet Size

Syntax: `AT+iMPS=n`

Limits the size of an outgoing PPP packet in dial-up environments. In effect, the MPS parameter limits the W24's MTU (Maximum Transfer Unit).

Parameters: `n=0..3`

Command Options:

`n=0` 1500 bytes

`n=1` 256 bytes

`n=2` 536 bytes

`n=3` 1024 bytes

Default: `n=0`

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

`AT+iMPS?` Returns current value followed by **I/OK**.

`AT+iMPS=?` Returns the message "0-3" followed by **I/OK**.

+iTTR - TCP Retransmit Timeout

Syntax: `AT+iTTR=n`

Sets the timeout, in milliseconds, after which an unacknowledged TCP packet will be retransmitted over a PPP connection by W24.

Parameters: `n=1000..65535`

Command Options:

Default: 3000 milliseconds

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

`AT+iTTR?` Reports the current value followed by **I/OK**.

`AT+iTTR=?` Returns the message "1000-65535" followed by **I/OK**.

+iBDRF - Define a Fixed Baud Rate on Host Connection

Syntax: AT+iBDRF=<n>

Sets the baud rate on host serial connection. This parameter is saved to nonvolatile memory and activated only after power-up.

Parameters: n=3..9|'a'|'h'

Command Options:

n=a Set baud rate to Auto Baud.

n=3 Set baud rate to 2400.

n=4 Set baud rate to 4800.

n=5 Set baud rate to 9600.

n=6 Set baud rate to 19200.

n=7 Set baud rate to 38400.

n=8 Set baud rate to 57600.

n=9 Set baud rate to 115200.

n=h Set baud rate to 230400.

When BDRF is set to a, W24 boots in auto baud rate mode. In this mode, W24 synchronizes on the first a or A character sent (normally as part of an AT or AT+i command) and detect its baud rate. The detected baud rate remains in effect until the W24 is power-cycled or issued the AT+iBDRA command.

If BDRF is set to a fixed value and the MSEL signal is pulled low for more than 5 seconds during runtime, W24 enters Rescue mode and forces auto baud rate detection. BDRF value will be used again upon the next power-up.

Default: 'a' (Auto Baud)

Result Code:

I/OK If *n* is within limits. W24 will continue operating in the current baud rate setting. Further power-ups will initialize the baud rate to the new selected value, until a different AT+iBDRF command is issued.

I/ERROR Otherwise.

AT+iBDRF? Returns the code for the specified fixed baud rate followed by **I/OK**.

AT+iBDRF=? Returns the message "3-9, 'a' or 'h'" followed by **I/OK**.

+iBDRM - Define a Fixed Baud Rate on W24<->Modem Connection

Syntax: AT+iBDRM=<*n*>

Sets the baud rate on modem connection. This parameter is saved to nonvolatile memory and activated after every power-up.

Parameters: *n*=3..9|'a'|'h'

Command Options:

n=a Set baud rate to Auto Baud.

n=3 Set baud rate to 2400.

n=4 Set baud rate to 4800.

n=5 Set baud rate to 9600.

n=6 Set baud rate to 19200.

n=7 Set baud rate to 38400.

n=8 Set baud rate to 57600.

n=h Set baud rate to 230400.

Default: 'a' (Auto Baud)

The W24<->modem connection will be set to the same baud rate as that detected on the host<->W24 connection.

Result Code:

I/OK If *n* is within limits. The W24 will continue operating in the current baud rate setting. Further power-up will initialize the baud rate to the new selected value, until a different AT+iBDRM command is issued.

I/ERROR Otherwise.

AT+iBDRM? Returns the code for the specified fixed modem baud rate followed by **I/OK**.

AT+iBDRM=? Returns the message "3-9, 'a' or 'h'" followed by **I/OK**.

+iBDRD - Baud Rate Divider

Syntax: AT+iBDRD=<*n*>

When set to '0', W24 sets its host USART baud rate according to the value of the BDRF parameter. When set to any value in the range 1-255, it divides the maximum supported baud rate - 3Mbps - by that value. The quotient of this division is set as the host baud rate, and the value of BDRF is ignored.

Parameters:

n=0 Host baud rate is determined by the BDRF parameter.

$n=1-255$	Host baud rate is set by dividing 3Mbps by n . For example, if $n=2$, the host baud rate will be set to 3Mbps-2=1.5Mbps.
Default:	0 (host baud rate taken from BDRF parameter).
Result Code:	
I/OK	If n is within limits.
I/ERROR	Otherwise.
AT+iBDRD?	Reports the current value followed by I/OK .
AT+iBDRD=?	Returns the message "0-255" followed by I/OK .

+iAWS - Activate WEB Server Automatically

Syntax:	AT+iAWS= v
	Sets Activate Web Server flag to v .
Parameters:	$v=0 \mid 1 \mid 2 \mid 3$
$v=0$	Automatic web server activation disabled.
$n>0$	Web server will be activated automatically when W24 goes online in SerialNET mode or as a result of a triggered Internet session initiation. Maximum number of concurrent browser connections is v .
Default:	0 (Automatic web server activation disabled).
Result Code:	
I/OK	if $v=0-3$.
I/ERROR	Otherwise.
AT+iAWS?	Reports the current value of the Activate WEB Server flag followed by I/OK .
AT+iAWS=?	Returns the message "0-3" followed by I/OK .

+iLATI - TCP/IP Listening Socket to Service Remote AT+i Commands

Syntax:	AT+iLATI=< $port$ >
	Sets the Remote AT+i service listening port number. When connected to the Internet, opens a TCP/IP listen socket on the local IP address and the specified $port$.
Parameters:	$port=0..65535$
Command Options:	
$port=0$	Remote AT+i service disabled.

<i>port=<portnum></i>	<p>Listening port to be used by a remote system when connecting to the W24 Family in order to send AT+i commands over the Internet.</p> <p>The listening socket will accept one remote connect request. When a remote system connects through the listen socket, W24 will disable its local host serial port and spawn a new TCP/IP socket, ready to receive AT+i commands. AT+i response strings will be transmitted back to the same socket.</p> <p>When the connected socket is closed, the local host serial port will be re-enabled and the listen socket will be ready to accept a new connection. The remote end may also issue the AT+iDOWN command to force W24 to disconnect and reboot.</p>
Default:	0 (Disabled).
Result Code:	
I/OK	Upon successfully opening the remote AT+i service TCP/IP listening socket.
I/ERROR	Otherwise.
AT+iLATI~n	Temporarily set the remote AT+i service Listen port. The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iLATI?	Returns current AT+i service listening port number followed by I/OK .
AT+iLATI=?	Returns the message "0-65535" followed by I/OK .

+iFLW - Set Flow Control Mode

Syntax: `AT+iFLW=n`

Sets the flow control mode.

Parameters: `n=0 .. 7`

Command Options:

- n*=
- | | |
|-------|--|
| Bit 0 | 0 = Host S/W flow control, using Wait/Continue control characters.
1 = Host hardware flow control based on ~CTS/~RTS hardware signals. |
| Bit 1 | 0 = No Modem flow control.
1 = Modem hardware flow control based on ~CTS/~RTS hardware signals. |
| Bit 2 | 0 = All hardware control signals: ~CTS, ~RTS, DTR and DSR are mirrored across W24 when transferring data transparently to the DCE.
1 = Hardware signal mirroring is disabled. |

Default: '0' (Host software flow control, no modem hardware flow control).

Result Code:

I/OK If *n* is within limits. See Note.

I/ERROR Otherwise.

`AT+iFLW~n` Temporarily set the flow control mode for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.

`AT+iFLW?` Returns current flow control mode followed by **I/OK**.

`AT+iFLW=?` Returns the message "0-7" followed by **I/OK**.

Note: When setting Bit 0 (Host hardware flow control), the ~CTSH signal must be LOW (enabled), otherwise W24 will return I/ERROR (063).

+iCPF - Active Communications Platform

Syntax: `AT+iCPF=n`

Sets the active communications platform to either modem or WLAN.

Parameters: `n=0 .. 1`

Command Options:

- n*=0 Sets active communications platform to dial-up or cellular modem. When the modem is online, any character, including <CR>, sent from the host that is not part of an AT+i command is transferred directly to the modem.

$n=1$ Sets active communications platform to WLAN.

Default: $n=1$ (WLAN)

Note: This default value is shipped from the factory. The AT+iFD command does not restore CPF to this value.

Result Code:

I/OK If n is within limits and the communications platform was actually changed.

I/ERROR Otherwise.

Followed by:

I/DONE After changing the current platform to modem. Allow a 2.5 sec. delay for W24 re-initialization.

-or-

I/ONLINE After changing the current platform to WLAN.

AT+iCPF= n Temporarily sets the active communications platform to n for one session. The permanent value will be restored after completing the next session, both if the session was successful or not. Note that I/ONLINE or I/DONE will be returned according to the new permanent communications platform.

AT+iCPF? Reports the currently active communications platform followed by **I/OK**.

AT+iCPF=? Returns the message "0-1" followed by **I/OK**.

+iPSE - Set Power Save Mode

Syntax: AT+iPSE= n

Enables or disables W24's Power Save Mode.

Parameters: $n=0$.. 255

Command Options:

$n=0$ Disables Power Save mode.

$n=1$..255 Enables Power Save mode. When Power Save mode is enabled, W24 automatically shuts down most of its circuits after a period of n seconds without any activity on the host or modem serial ports. Renewed activity on the serial ports restores W24 to full operational mode.

Default: 0 (Disabled).

Result Code:

I/OK If n is within limits.

I/ERROR Otherwise.

AT+iPSE? Reports the current Power Save mode setting followed by **I/OK**.

AT+iPSE=? Returns the message "0-255" followed by **I/OK**.

+iMRST - Turn the W24 Off

Syntax: AT+iMRST

The command initiates a W24 turn off, which switches the W24 to Off mode.

This command emulates the ON_N signal operation for turning the W24 off..

Result Code:

I/OK If command entered properly.

I/ERROR Otherwise.

+iS102 - Define Delay after Wakeup before Sending Data

The WKUPO_N signal is an active low W24 output that is set high by default. By asserting this signal low the host can be waked-up by W24. WKUPO_N alerts the host that W24 has data for the host (see [Figure 2-9](#)).

When W24 has no data for the host, WKUPO_N line must be de-asserted (set high).

The S102 and S100 commands configure the W24 sleep-mode behavior.

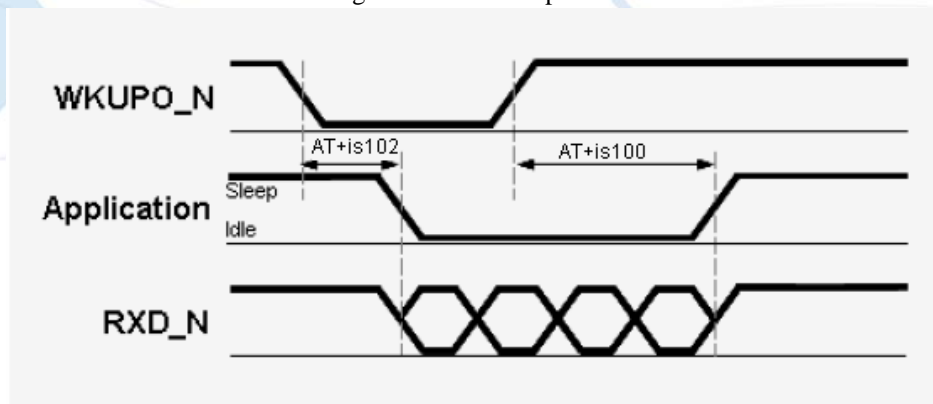


Figure 2-9: Lines Behavior During Wakeup Event

Syntax: AT+iS102=n

Defines the delay time, in milliseconds, that W24 waits when in SerialNET mode, after asserting the WKUPO_N signal, and before sending data on the host interface. This delay is required to allow the application enough time to re-activate from low power mode and switch to normal mode.

Parameters: n=0 .. 255

Default:	0 = The WKUPO_N signal and mechanism is disabled. In other words, W24 will never assert the WKUPO_N signal.
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iS102?	Returns current setting followed by I/OK .
AT+iS102=?	Returns the message "0-255" followed by I/OK .

+iS100 - Define Wait Interval Between Wakeup Events

Syntax:	AT+iS100= <i>n</i>
	Defines the minimal time interval, in milliseconds, that W24 waits before asserting WKUPO_N after de-asserting it. In other words, W24 will not assert the WKUPO_N signal if the time that had passed from the previous de-assertion of this signal is not at least the duration specified by the S100 parameter. This time interval is required to avoid frequent unnecessary wakeup events and consequent S102 delays. The S100 parameter is relevant only if S102>0, which enables the WKUPO_N signal operation.
Parameters:	<i>n</i> =0 .. 255
Default:	0
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iS100?	Returns current setting followed by I/OK .
AT+iS100=?	Returns the message "0-255" followed by I/OK .

+iSDM - Service Disabling Mode

Syntax:	AT+iSDM= <i>n</i>
	Sets the service disabling mode bits.
Parameters:	<i>n</i> =0 .. 7
Command Options:	

n= Bitmapped flags:
 Bit 0: Disable W24's response to ICMP ECHO (PING) requests. When this bit is set, W24 will not respond to any PING requests, thereby eliminating the possibility of a PING attack on W24.
 Bit 1: Disable W24's remote debug daemon. When this bit is set, W24 will not enable its internal (UDP) debug port, which is normally activated for administering remote support.
 Bit 3: Disable unauthenticated viewing of the W24's internal website. When this bit is set, the internal Web site may be browsed only if the remote browser provides the RPG parameter (password). In this case, when the RPG parameter contains a password value, W24's Configuration Web site will first display a password entry form. The remote end must submit the correct RPG value in order to continue to the Configuration site's home page. W24 uses the SHA1 hash algorithm throughout the authentication process, so actual password values are never transmitted. When this bit is set, but the RPG parameter is empty, the Configuration Web site is effectively disabled, as all password values will be rejected. However, if the RPG parameter contains the special '*' wildcard value, authentication is bypassed and the authentication form will be skipped altogether. In this case, the Configuration website's home page will be displayed immediately.

Default: 0 (All services enabled).

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iSDM? Returns current Service Disabling mode followed by **I/OK**.

AT+iSDM=? Returns the message "0-7" followed by **I/OK**.

+iDF - IP Protocol 'Don't Fragment' Bit Value

Syntax: AT+iDF=*n*

Sets the value of the Don't Fragment bit used in all subsequent IP packets.

Parameters: *n*=0 .. 1

Command Options:

n=0 IP packets transmitted may be fragmented by routers.

n=1 IP packets transmitted may not be fragmented by routers.

Default: 0

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iDF~ <i>n</i>	Temporarily sets the IP protocol Don't Fragment bit to <i>n</i> for one session. The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iDF?	Reports the current IP protocol Don't Fragment bit setting followed by I/OK .
AT+iDF=?	Returns the message "0-1" followed by I/OK .

+iCKSM - Checksum Mode

Syntax: AT+iCKSM=<*n*>

Sets W24's checksum mode. With this mode enabled, W24 calculates the checksum of data it returns to host upon receiving the AT+iSRCV command. At the same time, W24 expects the host to append checksum to the data it sends with the AT+iSSND command. W24 compares the checksum it calculates with the one calculated by the host to verify that data was not corrupted during transmission between host and W24.

Parameters:	<i>n</i> =0 1
Command Options:	
<i>n</i> =0	Checksum mode disabled.
<i>n</i> =1	Checksum mode enabled.
Default:	<i>n</i> =0 (checksum mode disabled).
Result Code:	
I/OK	If <i>n</i> is either '0' or '1'.
I/ERROR	Otherwise.

+iHIF - Host Interface

Syntax: AT+iHIF=*n*

Specifies the interface to be used for communication between the host processor and W24 in subsequent sessions. This parameter takes effect only after power-up.

Parameters:

<i>n</i> =0	Automatic host interface detection. In this mode, the first character sent from the host over one of the supported interfaces sets the host interface to be used throughout that session until the next W24 power cycle. If HIF is set to a fixed interface (<i>n</i> =1-5) and the MSEL signal is pulled low for more than 5 seconds during runtime, W24 switches to auto host interface detection mode (HIF=0).
-------------	---

<i>n</i> =1	USART0
<i>n</i> =2	USART1
<i>n</i> =3	USART2
<i>n</i> =4	USB Device
<i>n</i> =5	USB Host
Default:	0 (Automatic host interface detection).
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iHIF?	Reports the current value followed by I/OK .
AT+iHIF=?	Returns the message "0-5" followed by I/OK .

+iMIF - Modem Interface

Syntax:	AT+iMIF= <i>n</i>
	Specifies the interface to be used for communication between W24 and a dialup or cellular modem in subsequent sessions. This parameter takes effect only after power-up.
Parameters:	
<i>n</i> =1	USART0
<i>n</i> =2	USART1
<i>n</i> =3	USART2
<i>n</i> =4	USB Device
<i>n</i> =5	USB Host (only Motorola G24 USB GSM modem is supported).
Default:	2 (USART1).
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iMIF?	Reports the current value followed by I/OK .
AT+iMIF=?	Returns the message "1-5" followed by I/OK .

+iADCL - ADC Level

Syntax: AT+iADCL=<level>

Specifies an ADC base level, or threshold, in the range 0-255 that corresponds to an analog voltage measured on the input pin of W24's A/D converter.

Together with ADCD, these two parameters determine when the A/D converter asserts the GPIO pin specified by the ADCP parameter. ADCL must be greater than ADCD.

Parameters:

level=0 A/D converter polling is disabled.

level=1-255 ADC threshold level.

Default: 0 (polling disabled).

Result Code:

I/OK If *level* is within limits.

I/ERROR Otherwise.

AT+iADCL? Reports the current value followed by **I/OK**.

AT+iADCL=? Returns the message "0-255" followed by **I/OK**.

+iADCD - ADC Delta

Syntax: AT+iADCD=<delta>

Specifies an ADC delta. Together with ADCL, these two parameters determine when the A/D converter asserts the GPIO pin specified by the ADCP parameter. ADCD must be less than ADCL.

Parameters:

delta=0-255 A/D converter polling is disabled.

Default: 0 (zero delta).

Result Code:

I/OK If *delta* is within limits.

I/ERROR Otherwise.

AT+iADCD? Reports the current value followed by **I/OK**.

AT+iADCD=? Returns the message "0-255" followed by **I/OK**.

+iADCT - ADC Polling Time

Syntax: AT+iADCT=<interval>

Specifies the time interval between consecutive queries of the value of the A/D converter's register. W24's response time to value changes is up to 40ms.

Parameters:

interval=0 A/D converter polling is disabled.

interval=1-65535 Time interval, in milliseconds, between queries.

Default: 0 (polling disabled).

Result Code:

I/OK If *interval* is within limits.

I/ERROR Otherwise.

AT+iADCT? Reports the current value followed by **I/OK**.

AT+iADCT=? Returns the message "0-65535" followed by **I/OK**.

+iADCP - ADC GPIO Pin

Syntax: AT+iADCP=<pin>

Defines which of W24's general-purpose I/O pins (GPIO) is asserted by the A/D converter's polling mechanism.

Parameters:

pin=0 A/D converter polling is disabled.

pin=1-32 Pins 1-32 of PIOA (general-purpose I/O pins group A).

pin=33-64 Pins 1-32 of PIOB (general-purpose I/O pins group B).

pin=65-96 Pins 1-32 of PIOC (general-purpose I/O pins group C).

Default: 0 (polling disabled).

Result Code:

I/OK If *pin* is within limits.

I/ERROR Otherwise.

AT+iADCP? Reports the current value followed by **I/OK**.

AT+iADCP=? Returns the message "0-96" followed by **I/OK**.

+iRRA - W24 Readiness Report Activation

Syntax: AT+iRRA=<*n*>

Sets the type of W24 readiness indication sent to the host following a hardware reset.

Command Options::

- n*=0 No indication is sent.
- n*=1 An **I/ATI** message is sent, indicating W24 is ready to accept AT+i commands.
- n*=2 An **I/<IP Address>** message is sent, indicating W24 has an IP address and is ready for IP communication.
In a wireless LAN environment, this message indicates the following:
- W24 has established a connection with an AP.
 - W24 has completed WPA negotiations. (In case the WPA protocol is used, which means that the WLSI and WLPP parameters are not empty.)
 - W24 has been set to a static IP (DIP parameter is set to a value other than 0.0.0.0), or an IP address has been acquired from a DHCP server.
- In a LAN environment, this message indicates that W24 has been set to a static IP (DIP parameter is set to a value other than 0.0.0.0), or an IP address has been acquired from a DHCP server. In a dialup/cellular environment, this message indicates that a PPP connection has been successfully established with a PPP server.
- n*=3 The I/O pin specified by the RRHW parameter is asserted Low, indicating W24 is ready to accept AT+i commands.
- n*=4 The I/O pin specified by the RRHW parameter is asserted Low, indicating W24 has an IP address and is ready for IP communication.
- n*=5 An **I/ATI** message is sent, and the I/O pin specified by the RRHW parameter is asserted Low, indicating W24 is ready to accept AT+i commands.
- n*=6 An **I/<IP Address>** message is sent, and the I/O pin specified by the RRHW parameter is asserted Low, indicating W24 has an IP address and is ready for IP communication.

Default: 0 (No Indication).

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iRRA? Returns the current RRA value followed by **I/OK**.

AT+iRRA=? Returns the message "0-6" followed by **I/OK**.

Note:

- 1 The I/ATI and I/<IP Address> messages are sent only if:
 - W24 is set to communicate with the host over a fixed interface (HIF?0).
 - Either the host interface is not a USART, or host<->W24 baud rate is set to a fixed value (BDRF?a).
 - W24 is not configured to operate in SerialNET mode.
- 2 In a dialup/cellular environment, the I/<IP Address> message is sent only if W24 is configured to operate in Always Online mode (TUP=2).

+iRRHW - W24 Readiness Hardware Pin

Syntax: AT+iRRHW=<pin>

Defines which of W24's general-purpose I/O pins (GPIO) will be asserted Low to indicate W24 readiness to the host. W24 readiness indication is specified by the RRA parameter.

Parameters:

<i>pin</i> =0	No hardware indication is given.
<i>pin</i> =1-32	Pins 1-32 of PIOA (general-purpose I/O pins group A).
<i>pin</i> =33-64	Pins 1-32 of PIOB (general-purpose I/O pins group B).
<i>pin</i> =65-96	Pins 1-32 of PIOC (general-purpose I/O pins group C).
Default:	0 (no hardware indication is given).

Result Code:

I/OK If *pin* is within limits.

I/ERROR Otherwise.

AT+iRRHW? Reports the current value followed by **I/OK**.

AT+iRRHW=? Returns the message "0-96" followed by **I/OK**.

Note: Before specifying the I/O pin for this parameter, it is recommended that you consult the pin-out section of the W24 datasheet. Incorrect selection of pin might cause unexpected W24 behavior.

ISP Connection Parameters

+iISP*n* - Set ISP Phone Number

Syntax: `AT+iISPn=dial-s`

Sets the ISP's access phone numbers.

Use *n*=1 to set the ISP's primary access phone number.

Use *n*=2 to set the ISP's alternate number. The alternate number is dialed after exhausting all redial attempts of the primary number.

Parameters: *n*=1..2

dial-s= Telephone number string, composed of digits, ',', '-', 'W', 'w', '*', '#', '!' or ' '. See description of the standard ATD command.

Note: If a character that is defined as a delimiter is used within the dial string, the string must be entered between apostrophes.

Command Options:

dial-s='' Empty access number.

dial-s=<number> *number* will be set as ISP access number.

Default: Empty. No permanent ISP access number defined.

Result Code:

I/OK If *dial-s* is a legal phone number string.

I/ERROR Otherwise.

`AT+iISPn~dial-s` Temporarily sets the ISP's primary/alternate access number. The permanent value will be restored after completing the session, whether the session was successful or not.

`AT+iISPn?` Reports the current value of the ISP's primary/alternate access numbers. If the number does not exist, only <CRLF> is returned. The reply is followed by **I/OK**.

`AT+iISPn=?` Returns the message "Phone #" followed by **I/OK**.

+iATH - Set PPP Authentication Method

Syntax: `AT+iATH=v`

Sets authentication method to *v*.

Parameters: *v*=0..2

Command Options:

v=1 Use PAP authentication.

v=2 Use CHAP authentication.

Default: 1 (PAP).

Result Code:

I/OK If *v* is within limits.**I/ERROR** Otherwise.

AT+iATH~*v* Temporarily sets the authentication method to *v* for the duration of the next session. The permanent value will be restored after completing the session, whether the session was successful or not.

AT+iATH? Reports the current setting of the authentication method followed by **I/OK**.

AT+iATH=? Returns the message "0-2" followed by **I/OK**.

+iUSRN - Define Connection User Name

Syntax: AT+iUSRN=*user*

Sets connection user name.

Parameters: *user*=User name to be used when logging onto the ISP.

Command Options:

user= " " Empty: No user name defined.

user=<*user-name*> *user-name* is used to login to the ISP.

Default:

user= " " Empty. No user name defined. The login user name can be defined ad-hoc.

Result Code:

I/OK If *user* is an empty or legal ISP login name.**I/ERROR** Otherwise.

AT+iUSRN~*user* Temporarily sets the login user name to *user*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iUSRN? Reports the current login user name. If the user name does not exist, only <CRLF> is returned. The reply is followed by **I/OK**.

AT+iUSRN=? Returns the message "String" followed by **I/OK**.

+iPWD - Define Connection Password

Syntax: AT+iPWD=*pass*

Sets connection password.

Parameters: *pass*=Password to be used when logging onto the ISP.

Command Options:

<i>pass</i> ="	Empty: Nopassword defined.
<i>pass</i> =< <i>password</i> >	<i>password</i> is used to login to the ISP.
Default:	Empty. No password defined. The login password can be defined ad-hoc.

Result Code:

I/OK	If <i>password</i> is an empty or legal ISP login password.
I/ERROR	Otherwise.
AT+iPWD~ <i>pass</i>	Temporarily sets the login password to <i>pass</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iPWD?	Reports the current login password. The reported value will consist of '*' characters. The number of '*' characters shall reflect the number of characters in the actual password. If a password does not exist only <CRLF> will be returned. The reply is followed by I/OK .

AT+iPWD=? Returns the message "String" followed by **I/OK**.

+iRDL - Number of Times to Redial ISP

Syntax: AT+iRDL=*n*

Sets the number of times to redial ISP.

Parameters: *n*= Number of redial attempts to the ISP. If the ISP number is busy or the ISP does not pick up the line, the system will attempt to redial the ISP after a delay period as defined in the RTO parameter. If all redial attempts are exhausted, an attempt to dial the alternate ISP number will be made, if an alternate number exists. In the event that the number is busy or the ISP does not respond, the system will attempt to redial up to *n* times, as with the primary ISP number. If all redial attempts are exhausted, the system will quit with the error message: "All Redial Attempts Failed."

If the ISP does not pick-up the line, the W24 will timeout and determine a redial situation after the number of seconds stored in the WTC W24 parameter.

Command Options: *n*=0..20

Default: *n*=5.

Result Code:

I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.

AT+iRDL~ <i>n</i>	Temporarily sets the number of times to redial the ISP. The permanent number of redial attempts will be restored after completing the next session, whether the session was successful or not.
AT+iRDL?	Reports the current value of the number of times to redial ISP followed by I/OK .
AT+iRDL=?	Returns the message "0-20" followed by I/OK .

+iRTO - Delay Period between Redials to ISP

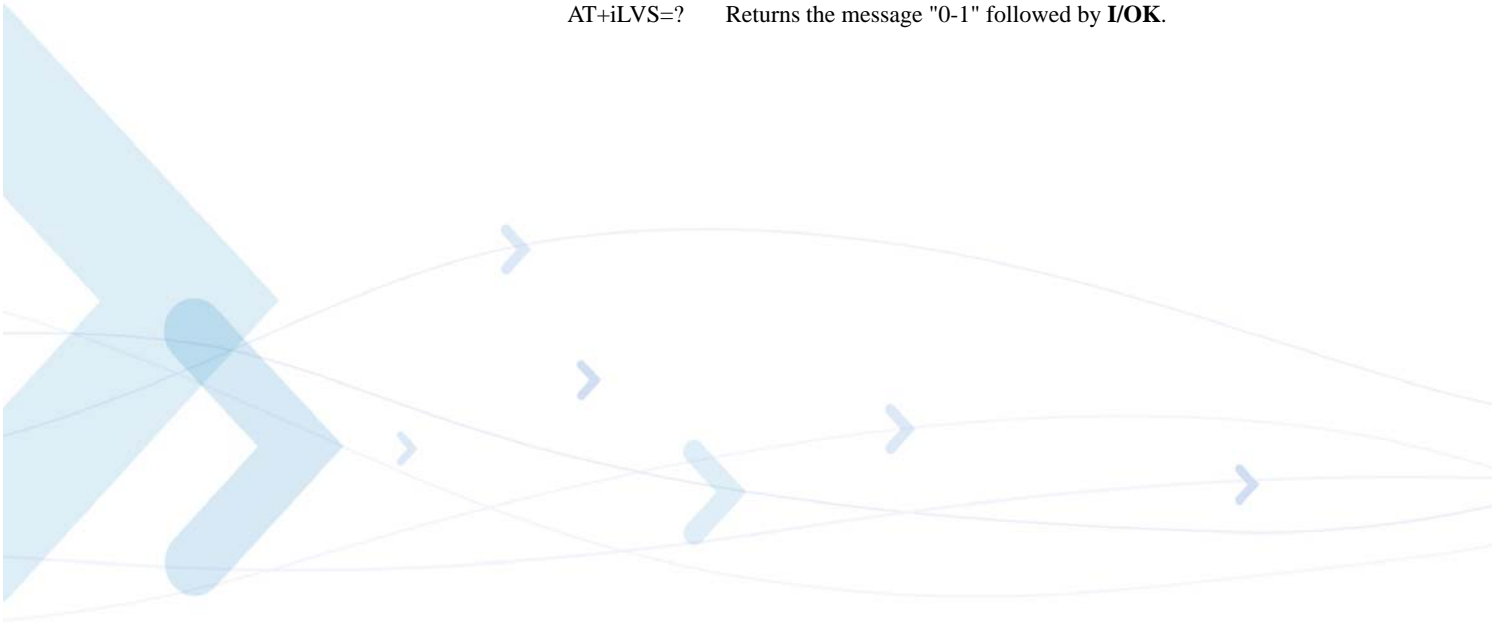
Syntax:	AT+iRTO= <i>n</i>
	Sets delay period, in seconds, between redials to ISP.
Parameters:	<i>n</i> = Number of seconds to delay before redialing the ISP, after a busy signal or in the event that the ISP did not answer the call. W24 will enforce a minimal 5 second delay for values of <i>n</i> less than 5 seconds.
Command Options:	<i>n</i> =0..3600 [seconds]
Default:	<i>n</i> = 180 [seconds].
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iRTO~ <i>n</i>	Temporarily sets the number of seconds to delay before redialing the ISP. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iRTO?	Reports the current number of seconds to delay before redialing the ISP. The reply is followed by I/OK .
AT+iRTO=?	Returns the message "0-3600" followed by I/OK .

Server Profile Parameters

+iLVS - 'Leave on Server' Flag

Syntax:	AT+iLVS= <i>v</i>
	Sets the 'Leave on Server' flag to <i>v</i> .
Parameters:	<i>v</i> = 0 1.
Command Options:	
<i>v</i> =0	After successful retrieval, messages will be deleted from server.

$v=1$	All messages will remain on server.
Default:	1
Result Code:	
I/OK	If $v = 0$ or 1.
I/ERROR	Otherwise.
AT+iLVS~ v	Temporarily sets the Leave on Server flag to v for the duration of the next session. The permanent value will be restored after completing the session, whether the session was successful or not.
AT+iLVS?	Reports the current value of the Leave on Server flag followed by I/OK .
AT+iLVS=?	Returns the message "0-1" followed by I/OK .



+iDNSn - Define Domain Name Server IP Address

Syntax: *AT+iDNSn[p]=IP*

Sets the Domain Name Server IP Address.

Use *n=1* to define the Primary IP address of the Domain Name Server associated with the ISP.

Use *n=2* to define the alternate IP address.

IP::=<nnn>.<nnn>.<nnn>.<nnn>

where,

<nnn>: [000..255].

Parameters:

n=1..2

p= Optional communication platform modifier for W24 Plus. Where, *p='S'* to force the (serial) dial-up platform and *p='L'* to force the WLAN platform. *p* may be used to select any platform. If *p* is omitted, the active platform will be used.

Command Options:

IP=0.0.0.0 Empty: No DNS defined.

IP=<IP add> *IP add.* Will be used to communicate to the Domain Name Server on the Internet.

Default: Empty. No DNS defined. The DNS must be defined ad-hoc. In a WLAN environment, an empty DNS (0.0.0.0) will acquire a value from the DHCP server (if DIP is 0.0.0.0). In a dial-up environment, the ISP will assign a DNS IP to an empty DNS, if the ISP supports RFC 1877 (PPP Extensions for Name Server Addresses).

Result Code:

I/OK If *IP* is an empty or legal IP address.

I/ERROR Otherwise.

AT+iDNSn[p]~IP Temporarily sets the DNS IP addresses. The permanent values will be restored after completing the next session, whether the session was successful or not.

AT+iDNSn[p]? Reports the current main/alternate DNS address. If no DNS address exists, 0.0.0.0 will be returned. The reply is followed by **I/OK**.

AT+iDNSn[p]=? Returns the message "IP Addr" followed by **I/OK**.

Note: This parameter may be omitted when the target server is defined with an IP addresses rather than a symbolic name.

+iSMTP - Define SMTP Server Name

Syntax: `AT+iSMTP[p]=server`

Permanently sets the SMTP Server Name or IP.

Parameters:

server An SMTP server name or IP address. Server names must be resolvable by the primary or alternate DNS.

p= Optional communication platform modifier for W24 Plus. Where, *p*='S' to force the (serial) dial-up platform and *p*='L' to force the LAN platform. *p* may be used to select any platform. If *p* is omitted, the active platform will be used.

Command Options:

server = " Empty: No server name defined.

server = <SMTP_SRVR> *SMTP_SRVR* will be used to locate and establish an SMTP connection when sending Email messages. If *SMTP_SRVR* is a symbolic name, a DNS server will be used to resolve the IP address.
Define +iSMA, +iSMU and +iSMP if the SMTP server requires authentication.

Default: Empty. No SMTP server defined. To send Email messages, the SMTP server name must be defined ad-hoc.
In a LAN environment, an empty SMTP server will acquire a value from the DHCP server (if DIP is 0.0.0.0).

Result Code:

I/OK If *server* is an empty or legal IP address or SMTP server address.

I/ERROR Otherwise.

`AT+iSMTP[p]~ server` Temporarily set the SMTP server name to *server*. The permanent server name will be restored after completing the next session, whether the session was successful or not.

`AT+iSMTP[p]?` Report the current SMTP server name. If a server name does not exist, only <CRLF> will be returned. The reply is followed by **I/OK**.

`AT+iSMTP[p]=?` Returns the message "String/IP" followed by **I/OK**.

+iSMA - SMTP Authentication Method

Syntax: AT+iSMA=*v*

Permanently sets SMTP authentication method.

Parameters: *v*= 0 or 1

Command Options:

v=0 SMTP authentication is disabled.

v=1 W24 will support the "AUTH LOGIN" SMTP authentication method, if forced by SMTP server.

Default: 0 (SMTP authentication disabled).

Result Code:

I/OK If *v* = 0 or 1.

I/ERROR Otherwise.

AT+iSMA? Report the current value of the SMTP authentication method. The reply is followed by **I/OK**.

AT+iSMA=? Returns the message "0-1".

+iSMU - Define SMTP Login User Name

Syntax: AT+iSMU=*user*

Permanently sets Authenticated SMTP login User Name.

Parameters: *user* = User Name to be used when logging on to an SMTP server that requires authentication (if SMA is set to a non zero value).

Command Options:

user="" Empty: No SMTP authentication User Name defined.

user=<*user-name*> *user-name* will be used to login to an authenticated SMTP server.

Default: Empty. No User Name defined.

Result Code:

I/OK If *user* is an empty or a legal SMTP login name.

I/ERROR Otherwise.

AT+iSMU~*user* Temporarily set the SMTP login User Name to *user*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iSMU? Report the current SMTP login User Name. If the User Name does not exist, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iSMU=? Returns the message 'String'. The reply is followed by **I/OK**.

+iSMP - Define SMTP Login Password

Syntax: AT+iSMP=*pass*

Permanently sets authenticated SMTP login.

Parameters: *pass* = Password to be used when logging on to an SMTP server that requires authentication.

Command Options:

pass='' Empty: No SMTP authentication password defined.

pass=<*password*> *password* will be used to login to an authenticated SMTP server.

Default: Empty. No password defined.

Result Code:

I/OK If *password* is an empty or a legal SMTP login password.

I/ERROR Otherwise.

AT+iSMP~*pass* Temporarily set the SMTP login password to *pass*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iSMP? Report the current SMTP login password. The reported value will consist of '*' characters. The number of '*' characters shall reflect the number of characters in the actual password. If a *password* does not exist, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iSMP=? Returns the message 'String'. The reply is followed by **I/OK**.

+iPOP3 - Define POP3 Server Name

Syntax: AT+iPOP3[*p*]=*server*

Permanently sets the POP3 Server Name or IP.

Parameters: *server* = a POP3 Server Name or IP address. The Server Name must be resolvable by the primary or alternate DNS.
p = optional communication platform modifier for W24 Plus. Where, *p*='S' to force the (serial) dial-up platform and *p*='L' to force the LAN platform. *p* may be used to select any platform. If *p* is omitted, the active platform will be used.

Command Options:

server='' Empty: No Server Name defined.

<i>server</i> = <POP3_SRVR>	POP3_SRVR will be used to locate and establish a POP3 connection when receiving Email messages. If POP3_SRVR is a symbolic name, a DNS server will be used to resolve the IP address.
Default:	Empty. No POP3 server defined. To retrieve Email messages, a POP3 Server Name must be defined ad-hoc. In a LAN environment, an empty POP3 server will acquire a value from the DHCP server (if DIP is 0.0.0.0).
Result Code:	
I/OK	If <i>server</i> is empty or a legal IP address or POP3 server name.
I/ERROR	Otherwise.
AT+iPOP3[p]~ <i>server</i>	Temporarily set the POP3 server name to <i>server</i> . The permanent server name will be restored after completing the next session, whether the session was successful or not.
AT+iPOP3[p]?	Report the current POP3 server name. If a server name does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iPOP3[p]=?	Returns the message 'String/IP'. The reply is followed by I/OK .

+iMBX - Define POP3 Mailbox Name

Syntax:	AT+iMBX= <i>mailbox</i>
	Permanently sets mailbox name.
Parameters:	<i>mailbox</i> = Mailbox name to be used for Email retrieve.
Command Options:	
<i>mailbox</i> = "	Empty: No mailbox Name defined.
<i>mailbox</i> = < <i>mbox-name</i> >	<i>mbox-name</i> will be used to retrieve Email messages.
Default:	Empty. No mailbox defined. To retrieve Email messages, a mailbox name must be defined ad-hoc.
Result Code:	
I/OK	If <i>mailbox</i> is empty or a legal mailbox name.
I/ERROR	Otherwise.
AT+iMBX~ <i>mailbox</i>	Temporarily set the mailbox name to <i>mailbox</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iMBX?	Report the current mailbox name. If a mailbox name does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iMBX=?	Returns the message 'String'. The reply is followed by I/OK .

+iMPWD - Define POP3 Mailbox Password

Syntax:	AT+iMPWD= <i>MBxPass</i>	Permanently sets POP3 mailbox password.
Parameters:	<i>MBxPass</i> = Mailbox password to be used for authentication, when retrieving Email messages from the mailbox.	
Command Options:		
	<i>MBxPass</i> = ""	Empty: No mailbox password defined.
	<i>MBxPass</i> = < <i>mbx-pass</i> >	<i>mbx-pass</i> will be used to authenticate receiver, when retrieving Email messages from the mailbox.
Default:		Empty. No mailbox password defined. To retrieve Email messages, the mailbox password must be defined ad-hoc.
Result Code:		
	I/OK	If <i>mbx-pass</i> is an empty or legal mailbox password.
	I/ERROR	Otherwise.
	AT+iMPWD~ <i>MbxPass</i>	Temporarily set the mailbox password to <i>MBxPass</i> . The permanent password will be restored after completing the next session, whether the session was successful or not.
	AT+iMPWD?	Report the current mailbox password. The reported value will consist of '*' characters. The number of '*' characters shall reflect the number of characters in the actual password. If a mailbox password does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
	AT+iMPWD=?	Returns the message 'String'. The reply is followed by I/OK .

+iINTSn - Define Network Time Server

Syntax:	AT+iINTSn=< <i>server</i> >	Sets the network time server name or IP.
Parameters:	<i>n</i> = 1..2 <i>server</i> = A network timeserver name or IP address. See "NIST Time Servers" for a list of NIST Time servers.	
Command Options:		
	<i>Server</i> = ""	Empty. No Network Time Server defined.
	<i>Server</i> = < <i>nts</i> >	The server name or IP address, nts, will be used to retrieve the current time-of-day - if the NTOD parameter is set to enable time-of-day retrieval. Current Time-of-Day will be returned in response to the RP8 command. Outgoing Email messages will be Time and Date stamped.

Default: Empty. No Network Time Servers defined.

Result Code:

I/OK

AT+iNTSn~server Temporarily sets the Network Time Server to value *server*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iNTSn? Reports the current value of NTS*n*. If NTS*n* is empty, an empty line containing only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iNTSn=? Returns the message 'String/IP Addr'. The reply is followed by **I/OK**.

+NTOD - Define Network Time-of-Day Activation Flag

Syntax: **AT+iNTOD=*n***

Sets the network time-of-day activation flag to *n*. If this flag is enabled, W24 will retrieve an updated time reading the next time it goes online.

Parameters: *n* = 0 or 1.

Command Options:

n=0 Network time retrieval from timeserver is disabled.

n=1 Network time retrieval is enabled - W24 will connect to the time server and retrieve an updated time reading each time it connects to the network. From that point on, W24 will maintain time internally. While W24 is online, network time will be refreshed every two hours.
Current time-of-day will be returned in response to the RP8 command. Outgoing e-mail messages will be time and date stamped.
The expiry data of an incoming server certificate in secure SSL communication will also be checked. If W24 cannot read the time from the time server, an SSL session cannot be established.

Default: 0 (time server retrieval disabled).

Result Code:

I/OK

AT+iNTOD~*n* Temporarily sets the network time-of-day activation flag to value *n*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iNTOD? Reports the current value of the network time-of-day activation flag followed by **I/OK**.

AT+iNTOD=? Returns the message '0-1'. The reply is followed by **I/OK**.

+iGMTO - Define Greenwich Mean Time Offset

Syntax:	AT+iGMTO= <i>n</i>
	Permanently sets W24 location's Greenwich mean time offset, in hours.
Parameters:	<i>n</i> = -12..12
Default:	0
Result Code:	
	I/OK
AT+iGMTO~ <i>n</i>	Temporarily set the Greenwich Mean Time Offset to value <i>n</i> . The permanent values will be restored after completing the next session, whether the session was successful or not.
AT+iGMTO?	Report the current value of GMTO. The reply is followed by I/OK .
AT+iGMTO=?	Returns the message '-12+12'. The reply is followed by I/OK .

+iDSTD - Define Daylight Savings Transition Rule

Syntax:	AT+iDSTD= <i>DST_rule</i>
	Permanently sets the daylight savings time transition rule.
Parameters:	<i>DST_rule</i> ::= "<HH1.DD1.MM1>;<HH2.DD2.MM2>"
	Where, <HH1.DD1.MM1> indicates the date when Daylight Saving Time starts and <HH2.DD2.MM2> indicates the date when Daylight Saving Time ends. HHn ::= Full Hour (two digits). DDn ::= Either specific day, or <F/L><Day of Week>. <F/L> ::= F = First, L = Last Day of the month. For example: FSun indicates the First Sunday of the month. <Day of Week> ::= {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"}. MMn ::= Month.
Command Options:	
<i>DST_rule</i> =""	Empty - no Daylight Saving Time definition is applied.
<i>DST_rule</i> =< <i>dst</i> >	Daylight Savings rule defined in <i>dst</i> will be applied to the time retrieved from the Time Server when reporting the current time.
Default:	Empty. No Daylight Saving Time is applied.
Result Code:	
	I/OK

AT+iDSTD~ <i>DST_rule</i>	Temporarily set the Daylight Saving Time Definition to <i>DST_rule</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iDSTD?	Report the current value of the Daylight Saving Time Definition. The reply is followed by I/OK .
AT+iDSTD=?	Returns the message 'String'. The reply is followed by I/OK .

+iPDS*n* - Define PING Destination Server

Syntax: AT+iPDS*n*=*Server*

Permanently sets the PING destination server name.

Parameters: *n* = 1..2
Server = A network server name or IP address.

Command Options:

Server='' Empty. No PING destination Server defined.

Server=<*nps*> The server name or IP address, *nps*, will be PING'ed in order to verify W24's online status, when W24 is in "Always Online" mode. If the primary server does not respond, W24 will try the secondary server (if it exists). When both servers do not respond to PING requests, W24 will retry to establish the connection by going offline and then online again.

Default: Empty. No PING destination Servers defined.

Result Code:
I/OK

AT+iPDS*n*~*Server* Temporarily set the PING destination server to value *Server*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iPDS*n*? Report the current value of PDS*n*. If PDS*n* is empty, an empty line containing only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iPDS*n*=? Returns the message 'String / IP Addr'. The reply is followed by **I/OK**.

+iPFR - PING Destination Server Polling Frequency

Syntax: AT+iPFR=*n*

Permanently sets the time interval, in seconds, upon which W24 will issue a PING request to one of the PING destination servers.

Parameters: *n* = 0..65535 [seconds].

Command Options:

Default:	0 (Disabled PING polling).
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iPFR~ <i>n</i>	Temporarily set the PING polling interval value for one session. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iPFR?	Report the current PING polling interval used. The reply is followed by I/OK .
AT+iPFR=?	Returns the message '0-65535'. The reply is followed by I/OK .

+iUFn - User Fields and Macro Substitution

Syntax:	AT+iUFn=<String>
	Permanently sets user field <i>n</i> .
Parameters:	<i>n</i> = 01..12 <i>String</i> = Parameter string-value.
Command Options:	
<i>String</i> =""	Empty User Field
<i>String</i> =<Str>	<i>Str</i> is stored in the specified User Field. Maximum <i>Str</i> length is 128 characters.
	A User Field may be used for general-purpose storage. In addition, a User Field may be used as a macro replacement wherever an AT+i Command <parameter> is allowed: The '#' character is used to prefix the UFn parameter to define indirection. When used, the value of the User Field will be substituted in the command before the command is processed. #UF01 -- #UF12 are allowed.
	For example: Given: AT+iUF01=ftp.domain.com Issuing: AT+iFOPN:#UF01:anonymous,myemail@domain.com Is equivalent to: AT+iFOPN:ftp.domain.com:anonymous,myemail@domain.com
	The advantage of this is that the FTP server may be specified dynamically by changing the UF01 parameter without requiring a change in the AT+iFOPN command.
Default:	Empty. No User Field value defined.
Result Code:	
I/OK	

AT+iUFn~<String>	Temporarily set User Field n to value String. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iUFn?	Report the current value of UFn. If the User Field is empty, an empty line containing only <CR/LF> will be returned. The reply is followed by I/OK .
AT+iUFn=?	Returns the message 'String'. The reply is followed by I/OK .

Email Format Parameters

+iXFH - Transfer Headers Flag

Syntax:	AT+iXFH=v
	Permanently sets 'Transfer Headers' flag to v.
Parameters:	v = 0 or 1
Command Options:	
v=0	Retrieve only Email body - No headers. BASE64 MIME attachments will be decoded by W24, on-the-fly.
v=1	Retrieve Email headers with Email body. Attachments shall not be decoded.
Default:	1
Result Code:	
I/OK	If v = 0 or 1.
I/ERROR	Otherwise.
AT+iXFH~v	Temporarily set the 'Transfer Headers Flag' to v for the duration of the next session. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iXFH?	Report the current value of the 'Transfer Headers Flag'. The reply is followed by I/OK .
AT+iXFH=?	Returns the message '0-1'. The reply is followed by I/OK .

+iHDL - Limit Number of Header Lines

Syntax:	AT+iHDL=n
	Sets maximum number of header lines to retrieve.
Parameters:	n = 0 - 255

Default:	0 (no limit).
Result Code:	
I/OK	If <i>n</i> is within limits.
I/ERROR	Otherwise.
AT+iHDL~ <i>n</i>	Temporarily set the maximum limit of header lines for the duration of the next session. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iHDL?	Report the current value of the header line limit. The reply is followed by I/OK .
AT+iHDL=?	Returns the message '0-255'. The reply is followed by I/OK .

+iFLS - Define Filter String

Syntax:	AT+iFLS= <i>str</i>
	Permanently sets a filter string.
Parameters:	<i>str</i> = ASCII string which qualifies an Email message to be listed or retrieved by the W24. This string must exist in the Email header for the message to qualify. If the string does not exist, the message will be ignored.
Command Options:	
<i>str</i> =""	Empty string: Filter disabled. All messages will be qualified for retrieval.
<i>str</i> =< <i>f/string</i> >	Set <i>f/string</i> to be the qualifying filter.
Default:	Empty. Filter disabled.
Result Code:	
I/OK	If <i>str</i> is an empty or legal filter string.
I/ERROR	Otherwise.
AT+iFLS~ <i>f/string</i>	Temporarily set the filter string to <i>f/string</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iFLS?	Report the current value of the filter string. If no filter is defined, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iFLS=?	Returns the message 'String'. The reply is followed by I/OK .

+iDELf - Email Delete Filter String

Syntax: *AT+iDELf=[#]*str**

Permanently sets the Email delete filter string.

Parameters: *str* = ASCII string which qualifies an Email message to be deleted from the mailbox. This string must exist in the Email header for the message to qualify. If the string exists in at least one header field, the message will be deleted from the mailbox during the next Email retrieve session (*AT+iRMM*).

Command Options:

str="" Empty string: delete filter disabled. No messages will be deleted.

str=<f/string> Set *f/string* to be the qualifying Email delete filter.

flag When the optional '#' (NOT) flag precedes the filter string, W24 will reverse the deletion criterion. In other words, W24 will delete all but Emails that qualify the filter

Default: Empty. Delete filter disabled.

Result Code:

I/OK If *str* is an empty or legal filter string.

I/ERROR Otherwise.

AT+iDELf~[#]f/string Temporarily set the Email delete filter string to *f/string*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iDELf? Report the current value of the Email delete filter string. If no filter is defined, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iDELf=? Returns the message 'String'. The reply is followed by **I/OK**.

+iSBJ - Email Subject Field

Syntax: *AT+iSBJ:subject*

Permanently sets Email header's Subject field.

Parameters: *subject* = Contents of subject field.

Command Options:

subject="" Empty string. 'Subject:' Field in Email header will be left empty.

subject=<subject string> The 'Subject:' field in the Email header will contain *subject string*.

Default: Empty.

Result Code:

I/OK If *subject* is an empty or legal string.

I/ERROR	Otherwise.
AT+iSBJ~<i>subject</i>	Temporarily set the contents of the 'Subject:' field of the next Email to be sent. The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iSBJ?	Report the current contents of the 'Subject:' parameter. If no subject is defined, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iSBJ=?	Returns the message 'String'. The reply is followed by I/OK .

+iTOA - Define Primary Addressee

Syntax:	AT+iTOA[n]=<i>Email@</i>
	Permanently sets Email addressee.
Parameters:	<i>Email@</i> = Email addressee. This is the default Email addressee, which will be used to direct Email messages sent by W24. <i>n</i> = optional index of addressee. When <i>n</i> is not specified, TOA00 (primary addressee) is used.
Command Options:	
<i>Email@</i> =""	Empty address: No addressee defined.
<i>Email@</i> =< <i>addr</i> >	<i>addr</i> will be used as a destination address for future Email SEND commands (+iEMA, +iEMB).
<i>n</i> =	01..50
Default:	Empty. No addressee defined.
Result Code:	I/OK
AT+iTOA[n]~<<i>add</i>>	Temporarily set the Email addressee to <i>add</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iTOA[n]?	Report the current value of the Email addressee. If the addressee does not exist, an empty line containing only <CRLF> will be returned. The reply is followed by I/OK .
AT+iTOA[n]=?	Returns the message 'String'. The reply is followed by I/OK .

+iTO - Email 'To' Description/Name

Syntax:	AT+iTO:<i>to</i>
	Permanently sets Email header's 'To:' description.
Parameters:	<i>to</i> = Contents of 'To:' description/name field.

Command Options:

to="" Empty string.

to=<*to_str*> The 'To:' description field in the Email header will contain *to_str*.

Default: Empty.

Result Code:

I/OK If *to* is an empty or legal string.

I/ERROR Otherwise.

AT+iTO~*to* Temporarily set the contents of the 'To:' description field of the next Email to be sent. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iTO? Report the current contents of the *to* parameter. If the *to* parameter is empty, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iTO=? Returns the message 'String'. The reply is followed by **I/OK**.

+iREA - Return Email Address

Syntax: AT+iREA=*Email*@

Permanently sets the Return Email Address. This is the Email address that will be used when replying to this Email.

Parameters: *Email*@ = Email addressee.

Command Options:

Email@="" Empty address: No return address defined.

Email@=<*addr*> *addr* will be used as the return Email address.

Default: Empty. No return Email address defined. The return Email address will be defined ad-hoc.

Result Code:

I/OK

AT+iREA~<*addr*> Temporarily set the return Email address to *addr*. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iREA? Report the current value of the return Email address. If the return Email address does not exist an empty line containing only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iREA=? Returns the message 'String'. The reply is followed by **I/OK**.

+iFRM - Email 'From' Description/Name

Syntax: *AT+iFRM:from*

Permanently sets Email header 'From:' description.

Parameters: *from* = Contents of 'From:' description field.

Command Options:

from="" Empty string.

from=<from string> The 'From:' description field in the Email header will contain *from string*.

Default: Empty.

Result Code:

I/OK If *from* is an empty or legal string.

I/ERROR Otherwise.

AT+iFRM~from Temporarily set the contents of the 'From:' description field of the next Email to be sent. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iFRM? Report the current contents of the *from* parameter. If the *from* parameter is empty, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iFRM=? Returns the message 'String'. The reply is followed by **I/OK**.

+iCCn - Define Alternate Addressee <n>

Syntax: *AT+iCCn=Email@*

Permanently sets alternative addressee.

Parameters: *n* = 1..4

Email@ = Email addressee. This is the Email address, which will be used to copy Email messages sent by the W24 to the primary addressee list.

Command Options:

Email@="" Empty address: Alternate addressee *n* not defined.

Email@=<addr> *addr* will be used as alternate Email addressee *n*.

Default: Empty. No alternate addressees defined.

Result Code:

I/OK

AT+iCCn~<addr>	Temporarily set alternate addressee <i>n</i> to <i>addr</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iCCn?	Report the current value of alternate addressee <i>n</i> . If the alternate addressee does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iCCn=?	Returns the message 'String'. The reply is followed by I/OK .

+iMT - Media Type Value

Syntax: AT+iMT=*type*

Permanently sets the media type used for generating Email messages with a MIME encapsulated attachment.

Parameters: *type* = Media type.

Command Options:

type=0..4 *type* will be used as the media type:
0 - text
1 - image
2 - audio
3 - video
4 - application.

Default: 4 (application).

Result Code:

I/OK If *type* is in the range: 0..4.

I/ERROR Otherwise.

AT+iMT~*type* Temporarily set the media type. The permanent value will be restored after completing the next session, whether the session was successful or not.

AT+iMT? Report the current media type value. The reply is followed by **I/OK**.

AT+iMT=? Returns the message '0-4'. The reply is followed by **I/OK**.

+iMST - Media Subtype String

Syntax: AT+iMST=*str*

Permanently sets the media subtype string used for generating Email messages with a MIME encapsulated attachment.

Parameters: *str* = Media subtype string.

Command Options:

<i>str</i> =""	Empty: No media subtype string defined, the default will be used.
<i>str</i> =< <i>string</i> >	<i>string</i> will be used as the media subtype string. A list of subtype strings is detailed in “MIME Content Types and Subtypes” on page A-1 .
Default:	'octet-stream'.
Result Code:	
I/OK	If <i>str</i> is an empty or a legal media subtype string.
I/ERROR	Otherwise.
AT+iMST~ <i>str</i>	Temporarily set the media subtype string to <i>str</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iMST?	Report the current media subtype string. If the string is empty, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iMST=?	Returns the message 'String'. The reply is followed by I/OK .

+iFN - Attachment File Name

Syntax:	AT+iFN= <i>fname</i>
	Permanently sets the attachment file name string used for generating Email messages with a MIME encapsulated attachment.
Parameters:	<i>fname</i> = Attachment file name.
Command Options:	
<i>fname</i> =""	Empty: No file name string defined, the default will be used.
<i>fname</i> =< <i>str</i> >	<i>str</i> will be used as the file name string when constructing a MIME attachment. The file name should be complete with an explicit extension.
Default:	W24 generated unique filename, without an extension.
Result Code:	
I/OK	If <i>fname</i> is an empty or a legal file name string.
I/ERROR	Otherwise.
AT+iFN~ <i>fname</i>	Temporarily set the file name string to <i>fname</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iFN?	Report the current file name string. If the filename is empty, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iFN=?	Returns the message 'String'. The reply is followed by I/OK .

HTTP Parameters

+iURL - Default URL Address

Syntax: `AT+iURL=URLadd`

Sets the URL address string used for downloading web pages and files and uploading files to web servers.

Parameters: `URLadd` = URL address string.

Command Options:

`URLadd=""` Empty: No URL address string defined.

`URLadd=<str>` *str* will be used as the URL address string when downloading a Web page or file.

The URL address format is:

<Protocol>://<host>[:<port>]/[<absolute_link>]/

Where,

<protocol> - HTTP or HTTPS

<host> - Web Server Name: IP address or server name resolved by DNS.

<port> - Port number on server. Default: 80 for HTTP, 443 for HTTPS.

<absolute link> - Absolute path name of Web page or file on server.

Default: None.

Result Code:

I/OK If *URLadd* is an empty or a legal URL address string.

I/ERROR Otherwise.

`AT+iURL~URLadd` Temporarily set the URL address string to *URLadd*. The permanent value will be restored after completing the next session, whether the session was successful or not.

`AT+iURL?` Report the current URL address string. If the URL address is empty, only <CRLF> will be returned. The reply is followed by **I/OK**.

`AT+iURL=?` Returns the message 'String'. The reply is followed by **I/OK**.

+iCTT - Define Content Type Field in POST Request

Syntax: `AT+iCTT=<string>`

Defines the contents of the "Content-type:" field that is sent in the POST request by the AT+iSLNK command. This field specifies the type of file being sent.

Parameters: `string`=max length 64 bytes.

Command Options:

<i>string</i> ="	Empty. A default value of "application/x-www-form-urlencoded" will be used, and the server will expect the data to be the data sent in a "Submit" of a form.
<i>string</i> =<Content-type>	Type of file being sent by the AT+iSLNK command.
Default:	Empty.
Result Code:	
I/OK	If <i>string</i> is empty or a legal string.
I/ERROR	Otherwise.

+iWPWD - Password for Application Website Authentication

Syntax:	AT+iWPWD= <i>Pass</i>	Permanently sets the application website's remote parameter update Password.
Parameters:	<i>Pass</i> = Password to be used for authentication, when accepting application Web site parameter updates from a remote Web browser.	
Command Options:		
<i>Pass</i> ="	Empty: Remote application Web site parameter updates over the Web are effectively disabled.	
<i>Pass</i> =< <i>password</i> >	<i>password</i> will be used to restrict application Web site parameter updates via a remote Web browser.	
<i>Pass</i> ="*"	A <i>password</i> will not be required to authenticate application Web site parameter updates via the Web, effectively unrestricting remote parameter updates.	
Default:	Empty. No Password defined. Application Web site parameter updates via a remote browser are fully restricted.	
Result Code:		
I/OK	If <i>pass</i> is an empty or legal Password.	
I/ERROR	Otherwise.	
AT+iWPWD~ <i>pass</i>	Temporarily set the application Web site parameter Update Password to <i>pass</i> . The permanent Password will be restored after completing the next session, whether the session was successful or not.	
AT+iWPWD?	Report the current Password. If a Password does not exist, only <CRLF> will be returned. The reply is followed by I/OK .	
AT+iWPWD=?	Returns the message 'String'. The reply is followed by I/OK .	

RAS Server Parameters

+iRAR - RAS RINGS

Syntax: `AT+iRAR=n`

Sets the number of RINGS that will activate W24's internal RAS if RAU is not empty.

Parameters: *n* = number of RINGS W24 will detect before answering an incoming call and activating its internal RAS.
If *n* is set to a value greater than 100 and an incoming call is picked up by the host or the modem after less than *n*-100 RINGS, W24 will activate its internal RAS.
The RAS server will negotiate a PPP connection if a '~' is received as the first character from the modem after the CONNECT line to indicate a PPP packet. Otherwise, W24 will revert to transparent mode communications, allowing the host to conduct direct modem to modem data transfer.

Command Options:

n= 2 .. 20.

+100 Add 100 to any RAR value to force W24 to activate its internal RAS even if the call was picked up by the host or the modem (if a '~' is received as the first character from the modem after the CONNECT line to indicate a PPP packet).

Default: *n* = 4.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iRAR? Returns RAR's current value. The reply is followed by **I/OK**.

AT+iRAR=? Returns the message '2-20'. The reply is followed by **I/OK**.

+iRAU - Define RAS Login User Name

Syntax: `AT+iRAU=user`

Permanently sets RAS login user name.

Parameters: *user* = User Name to be used for authentication when accepting a call from a PPP client connecting to W24's internal RAS.

Command Options:

user= Empty: W24's internal RAS is effectively disabled.

user=<*user-name*> *user-name* will be used to establish login rights of a remote PPP client connection to W24's internal RAS.

<i>user</i> ="*"	A user-name will not be required to authenticate a remote PPP client connection to W24's internal RAS, effectively unrestricted remote access.
Default:	Empty. W24's internal RAS is effectively disabled.
Result Code:	
I/OK	If <i>user</i> is an empty or a legal login User Name.
I/ERROR	Otherwise.
AT+iRAU~ <i>user</i>	Temporarily set the RAS login User Name to <i>user</i> . The permanent value will be restored after completing the next session, whether the session was successful or not.
AT+iRAU?	Report the current RAS login User Name. If the User Name does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iRAU=?	Returns the message 'String'. The reply is followed by I/OK .

+iRAP - Password for RAS Authentication

Syntax:	AT+iRAP= <i>Pass</i>
	Sets the RAS Password.
Parameters:	<i>Pass</i> = Password to be used for login authentication when accepting a call from a PPP client connecting to W24's internal RAS.
Command Options:	
<i>Pass</i> =" or <i>Pass</i> ="*"	A <i>password</i> will not be required to authenticate a remote PPP client connection to W24's internal RAS.
<i>Pass</i> =< <i>password</i> >	<i>password</i> will be used to restrict access of a remote PPP client connection to W24's internal RAS.
Default:	Empty. No Password defined.
Result Code:	
I/OK	If <i>pass</i> is an empty or legal Password.
I/ERROR	Otherwise.
AT+iRAP~ <i>pass</i>	Temporarily set the RAS password to <i>pass</i> . The permanent Password will be restored after completing the next session, whether the session was successful or not.
AT+iRAP?	Report the current RAS Password. If a Password does not exist, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iRAP=?	Returns the message 'String'. The reply is followed by I/OK .

LAN Parameters

+iMACA - MAC Address of W24

Syntax: *AT+iMACA=mac*

Permanently sets W24's MAC address.

Parameters: *mac* = MAC address. The MAC address may only be assigned **once** in the lifetime of the device, i.e., while the current MAC address is still FFFFFFFF. After a MAC address is assigned it cannot be changed or overwritten.

Command Options:

mac=<mac@> *mac@* must consist of 12 hexadecimal characters. If the current MAC is FFFFFFFF then *mac@* will become the permanent MAC address.

Default: MAC address is assigned at the factory.

Result Code:

I/OK If *mac* is a legal hexadecimal string, and the MAC address is being set for the first time.

I/ERROR Otherwise.

AT+iMACA? Report the current MAC address. If no MAC address has been defined, the reply will be "FFFFFFFF". The reply is followed by **I/OK**.

AT+iMACA=? Returns the message 'String'. The reply is followed by **I/OK**.

+iDIP - W24 Default IP Address

Syntax: *AT+iDIP=IP address*

Permanently sets W24's default IP address to *IP address*.

Parameters: *IP address* = IP address.

Command Options:

IP address = 0.0.0.0 Empty: At power-up, W24 will attempt to resolve an IP address via a DHCP server. The assigned address will be stored in the IPA (active IP address) parameter.

IP address = 255.255.255.255 *Reserved.*

IP address = <*IP ADDR*> *IP ADDR* will be assigned to W24. The address will be stored in the DIP parameter. The DIP parameter's value is copied into the IPA parameter after power-up and after the *AT+iDOWN* command.

Default:	Empty (0.0.0.0). No IPA defined. IP address will be resolved through a DHCP server, if one is available.
Result Code:	
I/OK	If <i>IP address</i> is empty or a legal IP address.
I/ERROR	Otherwise.
AT+iDIP?	Report the current Default IP address. The reply is followed by I/OK .
AT+iDIP=?	Returns the message 'IP addr'. The reply is followed by I/OK .

+iIPA - Active IP Address

Syntax:	AT+iIPA= <i>IP address</i>
	Changes the active IP to <i>IP address</i> .
Parameters:	<i>IP address</i> = IP address.
Command Options:	
<i>IP address</i> =< <i>IP ADDR..</i> >	<i>IP ADDR.</i> will be assigned as the active W24 IP address. Also changes the permanent Default IP address in nonvolatile memory. See description of the DIP parameter. Valid only for W24.
Default:	Contents of the DIP parameter at power up.
Result Code:	
I/OK	If <i>IP address</i> is empty or a legal IP address.
I/ERROR	Otherwise.
AT+iIPA~ <i>IP address</i>	Temporarily set the current IP address. The permanent IP address (stored in the DIP parameter) will be restored/resolved after completing the next session, whether the session was successful or not.
AT+iIPA?	Report the current IP address.
AT+iIPA=?	Returns the message 'IP addr'. The reply is followed by I/OK .

Note: When W24 is offline, the IP address is always 0.0.0.0.

+iIPG - IP Address of the Gateway

Syntax:	AT+iIPG= <i>IP address</i>
	Permanently sets the IP address of the gateway to be used by W24.
Parameters:	<i>IP address</i> = Gateway IP address.

Command Options:

IP address = 0.0.0.0 Empty: W24 will try to resolve the gateway IP address via DHCP, but **ONLY** if the DIP parameter value has been set to empty (0.0.0.0).

IP address = <*IP ADDR*.> *IP ADDR* will be used as the gateway IP address.

Default: Empty. No Gateway IP defined.

Result Code:

I/OK If *IP address* is empty or a legal IP.

I/ERROR Otherwise.

AT+iPG~IP address Temporarily set the gateway IP address. The permanent IP address will be restored/resolved after completing the next session, whether the session was successful or not.

AT+iPG? Report the current gateway IP. The reply is followed by **I/OK**.

AT+iPG=? Returns the message 'IP addr.'. The reply is followed by **I/OK**.

+iSNET - Subnet Address

Syntax: *AT+iSNET=IP mask*

 Sets the Sub Net to *IP mask*.

Parameters: *IP mask* = Subnet mask address.

Command Options:

IPmask = 0.0.0.0 Empty: W24 will try to resolve the subnet address via DHCP, but **ONLY** if the DIP parameter value has been set to empty.

IP mask = <*MASK*.> *MASK* will be used by W24 as the subnet mask.

Default: Empty. No subnet mask address defined.

Result Code:

I/OK If *IP mask* is empty or a legal IP mask.

I/ERROR Otherwise.

AT+iSNET~IP mask Temporarily set the subnet mask to *IP mask*. The permanent subnet mask will be restored/resolved after completing the next session, whether the session was successful or not.

AT+iSNET? Report the current subnet mask. The reply is followed by **I/OK**.

AT+iSNET=? Returns the message 'IP addr.'. The reply is followed by **I/OK**.

Wireless LAN Parameters

+iWLCH - Wireless LAN Communication Channel

Syntax: `AT+iWLCH=<channel>`

Sets the default WiFi communication channel.

When W24 is configured to operate in ad-hoc mode, this parameter must be given a value between 1 and 13 that defines the channel to be used for beacon transmission. When W24 joins an already existing ad-hoc network, it adopts that network's channel.

Parameters: `channel` = 0-13.

Default: 0 (Access Point).

Result Code:

I/OK If `channel` = 0-13.

I/ERROR Otherwise.

`AT+iWLCH?` Reports the currently configured WiFi communication channel followed by **I/OK**.

`AT+iWLCH=?` Returns the message '0-13'. The reply is followed by **I/OK**.

+iWLSI - Wireless LAN Service Set Identifier

Syntax: `AT+iWLSI=<ssid>`

Sets the destination Wireless LAN Service Set Identifier (SSID) string.

Parameters: `ssid` = SSID required for communications with a specific Access Point (AP). The AP must be configured with the same SSID.

Command Options:

`ssid=""` Empty. No SSID defined. W24 will communicate with any AP.

`ssid=<ID>` `ID` will be used as the destination SSID. `ID` must be configured in the AP for W24 to successfully communicate with that AP.

`ssid=*` Prevents W24 from automatically attempting to connect to an AP or ad-hoc network immediately after power-up. If the `ssid` parameter value is changed to (*) while W24 is already connected to an AP, the current connection will not be affected.

`ssid=!` Optional flag indicating ad-hoc mode. Upon power-up, W24 will continuously search for existing ad-hoc networks in its vicinity and join the one having the strongest signal.

<i>ssid</i> !=<ID>	W24 will search for an ad-hoc network with the specified <i>ID</i> . If it finds one it will join it, otherwise it will create a new network with this <i>ID</i> .
Default:	Empty. No SSID defined.
Result Code:	
I/OK	If <i>ssid</i> is an empty or legal SSID string.
I/ERROR	Otherwise.
AT+iWLSI~ <i>ssid</i>	Temporarily sets the SSID to <i>ssid</i> . The permanent value will be restored after completing the next session.
AT+iWLSI?	Reports the current <i>ssid</i> value followed by I/OK .
AT+iWLSI=?	Returns the message 'String'. The reply is followed by I/OK .

+iWLWM - Wireless LAN WEP Mode

Syntax:	AT+iWLWM= <i>md</i>
	Sets the Wireless LAN WEP operation mode.
Parameters:	<i>md</i> = 0..2.
Command Options:	
<i>md</i> =0	WEP Disabled.
<i>md</i> =1	WEP Enabled, using 64-bit keys.
<i>md</i> =2	WEP Enabled, using 128-bit keys.
Default:	0 - WEP disabled.
Result Code:	
I/OK	if <i>md</i> is within limits.
I/ERROR	Otherwise.
AT+iWLWM~ <i>md</i>	Temporarily set the WEP operation mode to <i>md</i> . The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iWLWM?	Report the current WEP mode used. The reply is followed by I/OK .
AT+iWLWM=?	Returns the message '0-2'. The reply is followed by I/OK .

+iWLKI - Wireless LAN Transmission WEP Key Index

Syntax:	AT+iWLKI= <i>ki</i>
	Sets the Wireless LAN transmission WEP-Key index.

Parameters:	<i>ki</i> = 1..4.
Command Options:	
<i>ki</i> =< <i>key_indx</i> >	When transmitting WiFi packets, the WEP key at position <i>key_indx</i> in the 4 key array will be used for packet encryption.
Default:	1.
Result Code:	
I/OK	If <i>ki</i> = 1..4.
I/ERROR	Otherwise.
AT+iWLKI~ <i>ki</i>	Temporarily set the transmission WEP key index to <i>ki</i> . The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iWLKI?	Report the current Wireless LAN transmission WEP key index. The reply is followed by I/OK .
AT+iWLKI=?	Returns the message '1-4'. The reply is followed by I/OK .

+iWLKn - Wireless LAN WEP Key Array

Syntax:	AT+iWLKn= <i>keyString</i>
	Permanently sets the Wireless LAN WEP keys in the 4-slot WEP key array.
Parameters:	<i>n</i> = 1..4. <i>keyString</i> = WEP key string represented by a Hexadecimal ASCII string.
Command Options:	
<i>keyString</i> =""	Empty: No WEP key defined in position <i>n</i> .
<i>keyString</i> =< <i>key</i> >	<i>key</i> will be used as the key string value in position <i>n</i> . The identical value must be configured in the same position in the AP router. <i>key</i> must be a Hexadecimal representation string, where each byte is described by 2 ASCII characters in the range ['0'..'9'], ['A'..'F'] or ['a'..'f']. When using 64-bit WEP (WLWM=1), <i>key</i> may contain up to 10 characters (defining 5 bytes). When using 128-bit WEP (WLWM=2), <i>key</i> may contain up to 26 characters (defining 13 bytes).
Default:	Empty. No WEP key defined.
Result Code:	
I/OK	If <i>keyString</i> is an empty or legal WEP key string.
I/ERROR	Otherwise.

AT+iWLKn~keyString	Temporarily set WEP key <i>n</i> to <i>keyString</i> . The permanent value will be restored after completing the next session, both if the session was successful or not.
AT+iWLKn?	Report the current WEP key value in position <i>n</i> . The reported value will consist of '*' characters. The number of '*' characters shall reflect the number of characters in the actual key string. If the key string is empty, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iWLKn=?	Returns the message 'String'. The reply is followed by I/OK .

+iWLPS - Wireless LAN Power Save

Syntax: AT+iWLPS=*n*

Sets a time interval during which the Marvell WiFi chipset connected to W24 remains in Power Save mode. Value changes take effect only after a SW or HW reset.

Parameters:

n=0 WiFi chipset Power Save mode is disabled.

n=1-5 The number of beacon periods during which the WiFi chipset remains in Power Save mode. The beacon period is set by the Access Point (AP) and is typically 100ms. In ad-hoc mode, the beacon period is set by the creator of the ad-hoc network - W24 - to 100ms.

Default: *n*=0 (Power Save mode disabled).

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iWLPS? Returns the current value stored in WLPS followed by **I/OK**.

AT+iWLPS=? Returns the message '0-5'. The reply is followed by **I/OK**.

+iWLPP - Personal Shared Key Pass-Phrase

Syntax: AT+iWLPP=<passphrase>

Sets the wireless LAN WPA-PSK pass-phrase.

Parameters: <passphrase> = Pass-phrase to be used in generating the WPA-PSK encryption key.

Command Options:

passphrase="" Empty - WPA security is disabled.

passphrase =<*pass*> If WLSI (SSID) is not empty, WPA-PSK security is enabled for WiFi connections and *pass* is used in generating the WPA-PSK encryption key. The allowed value for *pass* is an ASCII string containing 8-63 characters.

Default: Empty.

Result Code:

I/OK If *pass* is an empty or legal pass-phrase.

I/ERROR Otherwise.

AT+iWLPP~*passphrase* Temporarily set the wireless LAN WPA-PSK pass-phrase to *passphrase*.

AT+iWLPP? Report the current pass-phrase. The reported value consists of '*' characters. The number of '*' characters reflects the number of characters in the pass-phrase. If a pass-phrase is not defined, only <CRLF> are returned. The reply is followed by **I/OK**.

AT+iWLPP=? Returns the message 'String'. The reply is followed by **I/OK**.

+iWROM - Enable Roaming in WiFi

Syntax: AT+iWROM=<*n*>

Sets W24 to Roaming mode.

Parameters: *n*=0 | 1.

n=0 Disable Roaming mode.

n=1 Enable Roaming mode.

Default: *n*=0.

Result Code:

I/OK If *n* is a legal value.

I/ERROR Otherwise.

AT+iWROM? Returns the current WROM value followed by **I/OK**.

AT+iWROM=? Returns the message '0-1'. The reply is followed by **I/OK**.

+iWPSI - Periodic WiFi Scan Interval

Syntax: AT+iWPSI=*n*

Sets the time interval - *n* - between consecutive scans that W24 performs for APs in its vicinity.

Parameters: *n*=1-3600 seconds.

Default: *n*=5 seconds.

Result Code:

I/OK If n is a legal value.

I/ERROR Otherwise.

AT+iWPSI? Returns the current WPSI value followed by **I/OK**.

AT+iWPSI=? Returns the message '1-3600'. The reply is followed by **I/OK**.

+iWSRL - SNR Low Threshold

Syntax: AT+iWSRL=< n >

Sets a low SNR threshold for W24 in Roaming mode. If the SNR value of the signal from the AP that W24 is currently associated with drops below n , W24 is triggered by the SNR low event.

Parameters: $n=0-255$ dB.

Default: $n=10$ dB

Result Code:

I/OK If n is a legal value.

I/ERROR Otherwise.

AT+iWSRL? Returns the current WSRL value followed by **I/OK**.

AT+iWSRL=? Returns the message '0-255'. The reply is followed by **I/OK**.

+iWSRH - SNR High Threshold

Syntax: AT+iWSRH=< n >

Sets a high SNR threshold for W24 in Roaming mode. W24 will re-associate only with APs having SNR that is better than n .

Parameters: $n=0-255$ dB.

Default: 30 dB

Result Code:

I/OK If n is a legal value.

I/ERROR Otherwise.

AT+iWSRH? Returns the current WSRL value followed by **I/OK**.

AT+iWSRH=? Returns the message '0-255'. The reply is followed by **I/OK**.

+iWSIn - Wireless LAN Service Set Identifier Array

Syntax: AT+iWSI<n>=<ssid>

Sets the destination Wireless LAN Service Set Identifier (SSID) string into position *n* in the array. This array defines the order in which W24 attempts to connect to an AP or ad-hoc network.

Parameters:

n=0-9 *n*=0 is equivalent to the WLSI parameter and defines the default SSID. The default SSID (WSI0 or WLSI) determines the type of scanning that W24 performs. If the default SSID refers to an AP, all SSIDs on the list must be configured for APs as well. If the default SSID refers to an ad-hoc network (starts with the (!) character), all SSIDs on the list must be configured for ad-hoc networks as well (start with the (!) character).

The location of an SSID within the list defines its priority, where the first SSID has the top priority. The SSIDs must be configured consecutively. For example, if WSI0 and WSI2 are set but WSI1 is not, W24 ignores WSI2.

If, for example, W24 is connected to an AP having an SSID value defined by WSI3, and that SSID is set to a different value using the AT+iWSI3=*new SSID* command, the change will take effect immediately and W24 will attempt to associate with an AP having the new SSID. If, on the other hand, W24 is not currently connected to an AP with SSID defined by WSI3 and the value of WSI3 is changed, the change will take effect only upon the next connection attempt.

<ssid>=<ID> *ID* will be used as the destination SSID. *ID* must be configured in the AP for W24 to successfully communicate with that AP

Command Options: *The options below apply to WSI0 only.*

ssid='' Empty. No SSID defined. W24 will communicate with the strongest AP in its vicinity.

ssid=* Prevents W24 from automatically attempting to connect to an AP or ad-hoc network immediately after power-up. If the *ssid* parameter value is changed to (*) while W24 is already connected to an AP, the current connection will not be affected.

ssid=! Optional flag indicating ad-hoc mode. Upon power-up, W24 will continuously search for existing ad-hoc networks in its vicinity and join the one having the strongest signal.

Default: Empty. No SSID defined.

Result Code:

I/OK If *n* is a legal value.

I/ERROR Otherwise.

AT+iWSIn~*ssid* Temporarily sets the *n*th position in the array to *ssid*.

AT+iWSIn? Reports the current SSID value in position *n*.

AT+iWSIn=? Returns the message 'String'. The reply is followed by **I/OK**.

+iWPPn - Pre-Shared Key Passphrase Array

Syntax: `AT+iWPPn=<passphrase>`

Sets the Wireless LAN PSK passphrase for WPA and WPA2 encryption for each individual SSID in the array.

Parameters:

`n=0-9` 10 WPA passphrases, one for each SSID, respectively. Setting `WPP0=<passphrase>` is equivalent to setting the `WLPP` parameter, and vice versa.

`<passphrase>=<pass>` `pass` is the passphrase to be used in generating the PSK encryption key for WPA and WPA2. The allowed value for `pass` is an ASCII string containing 8-63 characters.

Command Options:

`<passphrase>=""` Empty - WPA security is disabled.

`<passphrase>=<pass>` If `WSIn` is not empty, `pass` is used in generating the PSK encryption key for `WSIn`.

Default: Empty.

Result Code:

I/OK If `n` is a legal value.

I/ERROR Otherwise.

» `AT+iWPPn~pass` Temporarily sets passphrase in the `n`th position to `pass`.

`AT+iWPPn?` Reports the current passphrase value in position `n`. The reported value consists of (*) characters. The number of (*) characters reflects the number of characters in the passphrase. If a passphrase is not defined, only `<CRLF>` is returned. The reply is followed by **I/OK**.

`AT+iWPPn=?` Returns the message 'String'. The reply is followed by **I/OK**.

+iWKYn - Wireless LAN WEP Key Array

Syntax: `AT+iWKYn=<KeyString>`

Sets the Wireless LAN WEP key for each individual SSID in the array.

Parameters:

`n=0-9` 10 WEP keys, one for each SSID, respectively. Setting `KeyString` with `n=0` is equivalent to setting `WLKI` and `WLK1-WLK4` parameters.

`<KeyString>` WEP key string represented by a hexadecimal ASCII string.

Command Options:

<i>KeyString</i> ="	Empty.
<i>KeyString</i> =< <i>key</i> >	<p><i>key</i> will be used as the <i>KeyString</i> value in position <i>n</i>. <i>key</i> must be a hexadecimal representation ASCII string, where each byte is described by two ASCII characters in the range [0..9], [A..F] or [a..f].</p> <p>When using 64-bit WEP encryption (WLWM=1), <i>key</i> can contain up to 10 characters (defining 5 bytes). When using 128-bit WEP encryption (WLWM=2), <i>key</i> can contain up to 26 characters (defining 13 bytes).</p>
Default:	Empty.
Result Code:	
I/OK	If <i>n</i> is a legal value.
I/ERROR	Otherwise.
AT+iWKY <i>n</i> = <i>key</i>	Temporarily sets WEP key in the <i>n</i> th position to <i>key</i> .
AT+iWKY <i>n</i> ?	Reports the current WEP key value in position <i>n</i> . The reported value consists of (*) characters. The number of (*) characters reflects the number of characters in the actual key string. If the key string is empty, only <CRLF> is returned. The reply is followed by I/OK .
AT+iWKY <i>n</i> =?	Returns the message 'String'. The reply is followed by I/OK .

+iWST*n* - Wireless LAN Security Type Array

Syntax: AT+iWST*n*=<*sec*>

Sets the Wireless LAN security type for each individual SSID in the array.

Setting WST0=<*sec*> is equivalent to setting the WLWM and WSEC parameters accordingly, and vice versa. For example, setting WST0=3 (WPA-TKIP) causes W24 to automatically set WSEC=0. Setting WST0=1 (WEP-64) automatically sets WLWM=1.

Parameters:

<i>n</i> =0-9	Index of SSID.
<i>sec</i> =0	No security.
<i>sec</i> =1	WEP-64.
<i>sec</i> =2	WEP-128.
<i>sec</i> =3	WPA-TKIP.
<i>sec</i> =4	WPA2-AES.

Default: 0

Result Code:

I/OK If *sec* is a legal value.

I/ERROR	Otherwise.
AT+iWSTn~<i>sec</i>	Temporarily sets security type of the n th position to <i>sec</i> .
AT+iWSTn?	Reports the current security type value in position n . The reply is followed by I/OK .
AT+iWSTn=?	Returns the message '0-4'. The reply is followed by I/OK .

+iWSEC - Wireless LAN WPA Security

Syntax: **AT+iWSEC= n**

Sets the WPA protocol type to be used for wireless LAN security. This parameter takes effect following either a hardware or software reset (AT+iDOWN) only. A change to this parameter during W24 operation does not affect the current connection.

Parameters:

n =0 WPA-TKIP protocol.

n =1 WPA2-AES protocol.

Default: 0

Result Code:

I/OK If n is within limits.

I/ERROR Otherwise.

AT+iWSEC? Reports the current value followed by **I/OK**.

AT+iWSEC=? Returns the message '0, 1'. The reply is followed by **I/OK**.

IP Registration Parameters

+iRRMA - IP Registration Mail Address

Syntax: **AT+iRRMA= *Email@***

Permanently sets the IP registration addressee.

Parameters: *Email@* = Email addressee. This addressee will receive a registration Email message after W24 establishes an Internet session connection as a result of an explicit AT+i command or as a result of automated Internet session establishment procedures. The Email will contain the W24's ID and dynamically assigned IP address, in ASCII form. See Email IP Registration.

Command Options:

Email@= " Empty address: No Email will be sent after W24 goes online.

Email@=<addr> *addr* will be used as the IP registration Email addressee.

Default: Empty.

Result Code:

I/OK

AT+iRRMA? Report the current value of the IP registration addressee. If the IP registration addressee does not exist, an empty line containing only <CR/LF> will be returned. The reply is followed by **I/OK**.

AT+iRRMA=? Returns the message 'String'. The reply is followed by **I/OK**.

+iRRSV - IP Registration Host Server Name

Syntax: AT+iRRSV=*server_name*:*port*

Permanently sets the IP registration server name or IP and port number to be used in an IP registration procedure.

Parameters: *server_name* = A server name or IP address. Server names must be resolvable by the primary or alternate DNS.
 port = 0..65535.

Command Options:

server_name='' Empty: No IP registration server name defined.

server_name=<*ip_registration_server*> *ip_registration_server* will be used to locate and establish a connection after W24 establishes an Internet session connection as a result of an explicit AT+i command or as a result of automated Internet session establishment procedures. The dynamically assigned IP address will be sent to the server in ASCII form, after which the socket will be closed. See Socket IP Registration.

port=<*port number*> It is assumed that the host server is "listening" on *port number*.

Default: Empty. No server defined.

Result Code:

I/OK If *ip_registration_server* is an empty or legal server name and *port* is within limits.

I/ERROR Otherwise.

AT+iRRSV? Report the current IP registration server name and port number. If a server name does not exist, only <CR/LF> will be returned. The reply is followed by **I/OK**.

AT+iRRSV=? Returns the message 'Name/IP:Port'. The reply is followed by **I/OK**.

+iRRWS - IP Registration Web Server

Syntax: *AT+iRRWS=url*

Permanently sets the IP registration web server URL.

Parameters: *url* = The web server URL to use for registration after going online.

Command Options:

url= " Empty: No IP registration URL defined.

url=<*Reg_URL*> *Reg_URL* will be used to dynamically register W24's IP and Port after going online as a result of an explicit AT+i command or as a result of automated Internet session establishment procedures. See Web Server IP Registration.

Default: Empty. No Registration Web server defined.

Result Code:

I/OK If *Reg_URL* is an empty or legal URL string.

I/ERROR Otherwise.

AT+iRRWS? Report the current IP registration Web server URL. If a URL does not exist only <CR/LF> will be returned. The reply is followed by **I/OK**.

AT+iRRWS=? Returns the message 'String'. The reply is followed by **I/OK**.

+iRRRL - IP Registration Return Link

Syntax: *AT+iRRRL=IP[:Port]*

Permanently sets the IP registration Return Link IP and Web Port.

Parameters: *IP* = IP address to use for registration after going online.
Port = Port number to assign to W24's Web server.
See description of RRRL when registering IP.

Command Options:

IP = 0.0.0.0 Empty: No Return Link defined.

IP = <*IP_addr*> *IP_addr* will be used when registering after establishing an Internet session, rather than the W24's actual local IP address. This is useful when the W24 receives an internal IP address behind a NAT. Assigning the NAT's IP address to *IP_addr* will allow reaching the W24 from the Internet. In SerialNET, the LPRT parameter may be pre-configured in the NAT to connect to the W24 device. See SerialNET Server Devices.

Port = *Web_port* Optional port to map W24's Web server in order to allow surfing W24 across a NAT in association with *IP_addr*.

Default: Empty. No return link IP and Port defined.

Result Code:

I/OK	If <i>IP</i> is a legal IP address and <i>Port</i> is a legal IP port number.
I/ERROR	Otherwise.
AT+iRRRL?	Report the current return link IP and port. The reply is followed by I/OK .
AT+iRRRL=?	Returns the message "Name/IP[:Port]". The reply is followed by I/OK .

+iHSTN - W24 Network Host Name

Syntax: AT+iHSTN=*host*

Permanently sets W24's Host Name.

Parameters: *host* = Symbolic Host Name string.

Command Options:

<i>host</i> =""	Empty: Do not attempt to register a symbolic host name. If the W24 is already registered in the DNS, the symbolic name will typically be cleared only after the last DHCP lease assigned to this W24 has expired.
<i>host</i> =<NAME>	NAME will be used to negotiate the registration of the W24 on the DNS server via the DHCP server. Host name negotiation will be implemented only during the next DHCP session. Typically this session will occur after a hardware reset or by issuing the AT+iDOWN command. Note that in order to achieve a successful host name registration, the W24 must utilize a DHCP (DIP = 0.0.0.0) and the DHCP server must both exist and be configured to dynamically add entries to the local DNS server. NAME will also be included in all IP registration method formats.

Default: Empty. No network host name defined.

Result Code:

I/OK	If <i>host</i> is empty or a string.
I/ERROR	Otherwise.
AT+iHSTN?	Report the current host name. The reply is followed by I/OK .
AT+iHSTN=?	Returns the message "String". The reply is followed by I/OK .

SerialNET Mode Parameters

+iHSRV | +iHSRn - Host Server Name/IP

Syntax: `AT+i{HSRV | HSRn} = server_name:port`

Sets the host server-name or IP and port number to be used in SerialNET mode.

Use *n*=0 or HSRV to define the primary server.

Use *n*=1 or 2 to define secondary servers.

Parameters: *n* = 0 .. 2

server_name = A server name or IP address. Server names must be resolvable by the primary or alternate DNS.

port = 0..65535.

Command Options:

server_name=""

Empty: No server name defined. Serial data transmitted from device in SerialNET mode will be ignored until a remote client accesses W24.

server_name=<server>

server will be used in SerialNET mode to locate and establish a connection when serial data is transmitted from the device or when "Auto Link" SerialNET modes are defined. The server name may be any legal Internet server name, which can be resolved by the W24's DNS (Domain Name Server) settings. The server name may also be specified as an absolute IP address given in DOT form. If the primary server does not respond, W24 will try the secondary servers (if they are defined).

port=<port number>

It is assumed that the host server is "listening" on *port number*.

Default: Empty.

Result Code:

I/OK

If *server_name* is an empty or legal server name and *port* is within limits.

I/ERROR

Otherwise.

`AT+i{HSRV | HSRn}?`

Report the current host server and port as: <*server*>:<*port*>. If a server name does not exist, only <CRLF> will be returned. The reply is followed by **I/OK**.

`AT+i{HSRV | }HSRn=?`

Returns the message 'Name/IP:Port'. The reply is followed by **I/OK**.

+iHSS - Assign Special Characters to Hosts

Syntax: `AT+iHSS= <control_characters>`

When W24 is connected to HSR_n (where $n=0..2$) in SerialNet mode, and character <C_m> (where HSS=<C1><C2><C3>) arrives from the host, W24 will close the socket to remote server HSR_n, flush all characters received from host prior to <C_m>, and open a socket to remote server HSR_m. In the special case when $n=m$, W24 doesn't do anything. In any case, the control character will not be sent to remote server over the socket. W24 doesn't perform software reset, and stores all characters received from the host in MBTB (if defined). In addition, the SNRD parameter doesn't have any affect.

Parameters: `control_characters` = A string containing three control characters.

Command Options:

`control_characters=""` No control characters are defined. W24 will not respond to control characters to switch among HSRVs.

`control_characters=<string>` `string` is <C0><C1><C2>, where <C_i> is an ASCII character or a binary escape sequence (or an empty character). A binary escape sequence is represented as \xhh (4 characters) where h is a hexadecimal digit 0..9 or A..F. For example: `AT+iHSS="abc"`.

Default: Empty.

Result Code:

I/OK If `control_characters` is an empty or legal string.

I/ERROR Otherwise.

Example: `at+ihss=\x23\x24\x00`

When a number sign character '#' is received from host (ASCII 023 in hexadecimal notation), switch to primary remote server (HSR0). When a dollar sign '\$' arrives, switch to HSR1. When a Null character arrives, switch to HSR2.

+iDSTR - Define Disconnection String for SerialNET Mode

Syntax: `AT+i[!]DSTR:<disconnect_string>`

Permanently sets SerialNET device disconnection string. In a modem environment, W24 also goes offline following this event.

Parameters: `disconnect_string` = The string expected on a serial link to signal socket disconnection.

Command Options:

`disconnect_string= "` Empty string - the connection will never be terminated due to a string arriving on serial link.

disconnect_string=<*string*> *string* received on serial link signals socket disconnection.
string consists any combination of printable ASCII characters and characters represented by two hexadecimal digits, such as: \xhh, where h is a hexadecimal digit 0..9 or A..F. Hexadecimal representation allows specifying non-printable characters.

! W24 will not send a DSTR to the socket upon detection. When this flag is not specified, W24 will send a DSTR each time it detects it.

Default: Empty.

Result Code:

I/OK If *disconnect_string* is an empty or legal string.

I/ERROR Otherwise.

AT+iDSTR? Reports the current contents of the *disconnect_string* parameter. If the *disconnect_string* parameter is empty, only <CRLF> are returned. If the '!' flag is specified, the " *" string is appended to the report.
For example, the reply to a AT+iDSTR? command will be "&&&*" in case AT+!DSTR=&&& was previously specified.
The reply is followed by **I/OK**.

AT+iDSTR=? Returns the message 'String' followed by **I/OK**.

+iLPRT - SerialNET Device Listening Port

Syntax: AT+iLPRT=*n*

Permanently sets the port number on which W24 will listen for client connections in SerialNET mode.

Parameters: *n* = 0-65535.

Default: 0 (no port).

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iLPRT? Report the current value of the SerialNET device listen port.
The reply is followed by **I/OK**.

AT+iLPRT=? Returns the message "0-65535". The reply is followed by **I/OK**.

+iMBTB - Max Bytes To Buffer

Syntax: AT+iMBTB=*n*

Permanently sets max bytes to buffer while the W24 is establishing an Internet connection.

Parameters:	n = number of bytes to buffer while establishing the connection in SerialNET mode.
Command Options:	$n = 0 \dots 2048$
Default:	0 - No Buffering.
Result Code:	
I/OK	If n is within limits.
I/ERROR	Otherwise.
AT+iMBTB?	Report the current setting of max bytes to buffer. The reply is followed by I/OK .
AT+iMBTB=?	Returns the message "0-2048". The reply is followed by I/OK .

+iMTTF - Max Timeout to Socket Flush

Syntax:	AT+iMTTF= n Sets max inactivity timeout before flushing the SerialNET socket.
Parameters:	n = number of milliseconds of inactivity on serial link to signal socket flush in SerialNET mode.
Command Options:	$n = 0 \dots 65535$
Default:	0 - No timeout.
Result Code:	
I/OK	If n is within limits.
I/ERROR	Otherwise.
AT+iMTTF?	Report the current timeout before SerialNET socket flush in milliseconds. The reply is followed by I/OK .
AT+iMTTF=?	Returns the message "0-65535". The reply is followed by I/OK .

+iFCHR - Flush Character

Syntax:	AT+iFCHR= <i>flush_chr</i> Permanently sets flush character in SerialNET mode.
Parameters:	<i>flush_chr</i> = character received on serial link to signal socket flush in SerialNET mode.
Command Options:	

flush_chr = " Empty: No Flush character defined. The SerialNET socket will not be flushed as a result of receiving a special flush character.

flush_chr = 'a' - 'z' | 'A' - 'Z' | '0' - '9' | <hex_char>
where,
<hex_char>= \x<hh>
<hh> = 00-FF

Default: Empty. No flush character defined.

Result Code:

I/OK If *flush_chr* is empty or a legal character representation.

I/ERROR Otherwise.

AT+iFCHR? Report the current flush character. The reply is followed by **I/OK**.

AT+iFCHR=? Returns the message "String". The reply is followed by **I/OK**.

+iMCBF - Maximum Characters before Socket Flush

Syntax: AT+iMCBF=*n*

Permanently sets max number of characters before flushing the SerialNET socket.

Parameters: *n* = maximum number of characters received on the serial link before flushing the SerialNET socket.

Command Options:

n = 0 .. 1460

Default: 0 - No specific limit. Flushing governed by Network.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iMCBF? Report the current maximum number of characters before flushing the SerialNET socket. The reply is followed by **I/OK**.

AT+iMCBF=? Returns the message "0-1460". The reply is followed by **I/OK**.

+iIATO - Inactivity Timeout

Syntax: AT+iIATO=*n*

Permanently sets maximum inactivity timeout in seconds to signal socket disconnection in SerialNET mode. When signaled, W24 will close the connected SerialNET communication socket. In a modem environment, the W24 will also go offline following this event.

When W24 is in iRouter mode and TUP< >2, if no activity is detected for the specified period, W24 will disconnect its modem side and go offline.

Parameters: *n* = number of seconds of inactivity, on a connected SerialNET socket, to signal socket disconnection.
In iRouter mode, this number specifies a period of no activity on either the WiFi or modem/cellular side.

Command Options:

n = 0 .. 65535

When W24 is in Server SerialNET mode (LPRT defined) and it goes online in response to a triggering event: RING signal, MSEL signal pulled low or AT+I!SNMD -- timeout calculation commences only after the W24 opens the Listen port. When the Web server is activated (using AWS=1), an external reference to the Web server will restart the IATO timeout calculation.

Default: 0 - No timeout limit.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iIATO? Report the current inactivity timeout in seconds to signal socket disconnection in SerialNET mode. The reply is followed by **I/OK**.

AT+iIATO=? Returns the message "0-65535". The reply is followed by **I/OK**.

+iSNSI - SerialNET Device Serial Interface

Syntax: `AT+iSNSI=settings_str`

Sets serial interface settings for SerialNET mode.

Parameters: `settings_str` = Serial link settings in SerialNET mode.

Command Options:

`settings_str="<baud>,<data_bits>,<parity>,<stop_bits>,<flow>"`

where,

`<baud>` = 0..9 or h

`<data_bits>` = 7 | 8

`<parity>` = N | E | O

`<stop_bits>` = 1

`<flow>` = 0 | 1

The following table summarizes supported baud rates.

Default: "5,8,N,1,0" - baud rate 9600bps, 8 bits, No parity, 1 stop bit, no flow control.

Result Code:

I/OK If `settings_str` is a valid serial link setting string.

I/ERROR Otherwise.

`AT+iSNSI?` Reports the current serial settings string. The reply is followed by **I/OK**.

`AT+iSNSI=?` Returns the message "String". The reply is followed by **I/OK**.

The table below summarizes the supported bad rates.

Baud Code	Baud Rate	Baud Code	Baud Rate
0	See Note below.	6	19,200
1	600	7	38,400
2	1200	8	57,600
3	2400	9	115,200
4	4800	h	230,400
5	9600		

Note: Baud Code '0' means that host<->W24 baud rate in SerialNET mode is determined according to the value of the BDRD parameter.

+iSTYP - SerialNET Device Socket Type

Syntax: AT+iSTYP=*v*

Sets SerialNET socket type to *v*.

Parameters: *v* = 0 or 1.

Command Options:

v=0 TCP

v=1 UDP

Default: 0 (TCP).

Result Code:

I/OK If *v* = 0 or 1.

I/ERROR Otherwise.

AT+iSTYP? Reports the current value of the SerialNET socket type. The reply is followed by **I/OK**.

AT+iSTYP=? Returns the message "0-1". The reply is followed by **I/OK**.

+iSNRD - SerialNET Device Re-Initialization Delay

Syntax: AT+iSNRD=*n*

Sets SerialNET mode re-initialization delay in seconds.

Parameters: *n* = number of seconds W24 will pause before re-initializing SerialNET mode after a failed attempt to establish a socket connection to the peer or a connection related fatal error. A new SerialNET connection will only be attempted after SerialNET re-initializes. The SNRD delay will not be in effect as a result of an Escape Sequence ('+++').

Command Options:

n = 0..3600

Default: 0 - No delay.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iSNRD? Report the current SerialNET re-initialization delay in seconds. The reply is followed by **I/OK**.

AT+iSNRD=? Returns the message "0-3600". The reply is followed by **I/OK**.

+iSPN - SerialNET Server Phone Number

Syntax: *AT+iSPN=number*

Permanently sets the SerialNET phone number to use to wake up a remote SerialNET server.

Parameters: *number* = Telephone number to use to dial up a remote SerialNET server in order to wake it up and activate its preprogrammed Ring-Response procedures. The SerialNET client will attempt RDL redials. During each dial-up attempt it will wait for SDT seconds before hanging up.

Command Options:

number = Telephone number string, composed of digits, ',', '-', 'W', 'w', '*', '#', '!' or ' '. See description of the standard ATD command.

Default: Empty. Do not attempt to wake up a remote SerialNET server.

Result Code:

I/OK If *number* is a legal phone number string.

I/ERROR Otherwise.

AT+iSPN? Report the current SerialNET wakeup telephone number. The reply is followed by **I/OK**.

AT+iSPN=? Returns the message "Phone #". The reply is followed by **I/OK**.

Note: If a character that is defined as a Delimiter is used within the dial string, the string must be entered between apostrophes.

+iSDT - SerialNET Dialup Timeout

Syntax: AT+iSDT=*n*

Permanently sets the SerialNET Dial timeout when waking up a remote SerialNET server.

Parameters: *n* = Number of seconds to allow after dialing up the remote SerialNET server, before hanging up.

Command Options:

n = 0..255 [seconds].

Default: 20 [seconds].

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iSDT? Report the current SerialNET dial timeout. The reply is followed by **I/OK**.

AT+iSDT=? Returns the message "0-255". The reply is followed by **I/OK**.

+iSWT - SerialNET Wake-Up Timeout

Syntax: AT+iSWT=*n*

Sets the SerialNET wake-up timeout when waking up a remote SerialNET server.

Parameters: *n* = Number of seconds to allow the entire SerialNET server wakeup procedure before hanging up and retrying.

Command Options:

n = 0..65535 [seconds].

Default: 600 [seconds].

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iSWT? Report the current SerialNET Wake-up timeout. The reply is followed by **I/OK**.

AT+iSWT=? Returns the message "0-65535". The reply is followed by **I/OK**.

+iPTD - SerialNET Packets to Discard

Syntax: AT+iPTD=*n*

Sets the number of packets to be cyclically discarded in a SerialNET mode session. A packet is defined as the group of characters received on the serial link, meeting one (or more) of the socket flush conditions defined (+iFCHR, +iMTTF, +iMCBF).

Parameters: *n* = 0 - 65535.

Default: 0 - No packet filtering. All data is transferred.

Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

AT+iPTD? Report the current value. The reply is followed by **I/OK**.

AT+iPTD=? Returns the message "1-65535". The reply is followed by **I/OK**.

Remote Firmware Update Parameters

+iUEN - Remote Firmware Update Flag

Syntax: AT+iUEN=<*v*>

Sets the remote firmware update flag.

Parameters: *v* = 0 or 1.

Command Options:

v=0 Update only to a firmware version that is newer than the currently installed one.

v=1 Update to any firmware version available.

Default: 0

Result Code:

I/OK If *v* = 0 or 1.

I/ERROR Otherwise.

AT+iUEN~*v* Temporarily set the remote firmware update flag to *v* for the duration of the current session. The permanent value will be restored after completing the current session.

AT+iUEN? Reports the current value of the remote firmware update flag. The reply is followed by **I/OK**.

AT+iUEN=? Returns the message "0-1". The reply is followed by **I/OK**.

+iUSRV - Remote Firmware Update Server Name

Syntax: `AT+iUSRV="<protocol>://<host>[:<port>][<relative_path>]"`

Sets name of server to be used for updating W24 firmware remotely. This server must contain one or more firmware .imz files. The actual update process is initiated using the AT+iRFU command.

Parameters: `<protocol>` = http or ftp
`<host>` = Host name or IP address
`<port>` = 1..65535
 Default port for http is 80. Default port for ftp is 21.
`<relative_path>` = Path to a directory which contains one or more .imz files on the host or a path to a text file containing a list of one or more <CRLF>-separated .imz filenames. relative_path must be relative to the FTP home directory. If relative_path contains sub-directories, they can be divided using either '\' or '/'. absolute_path must end with '\' or '/'.

Command Options:

AT+iUSRV= Empty. No server name defined.

Default: Empty. No dedicated remote firmware update server defined.

Result Code:

I/OK If *host* is an empty or legal host name.

I/ERROR Otherwise.

AT+iUSRV~
 "<protocol>://<host>" Temporarily set the firmware update server name to *host*. The permanent value will be restored after completing the next session.

AT+iUSRV? Report the current firmware update server name. If a server name is not defined, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iUSRV=? Returns the message 'String / IP Addr'. The reply is followed by **I/OK**.

Example: `at+iusrv="ftp://172.20.101.5:21/RFU_CO2128/"`

+iUUSR - Remote Firmware Update FTP User Name

Syntax: AT+iUUSR=<username>

Sets name of user to logon to the FTP server defined in the AT+iUSRV parameter.

Parameters: <username> = Name of user to logon to the FTP server. This must be a registered user on the FTP server. Some servers allow anonymous login, in which case *username*=anonymous.

Command Options:

AT+iUUSR=" Empty. No user name defined.

Default: Empty. No user name defined.

Result Code:

I/OK If *username* is an empty or legal user name.

I/ERROR Otherwise.

AT+iUUSR~<username> Temporarily set the user name to *username*. The permanent value will be restored after completing the next session.

AT+iUUSR? Report the current user name. If a user name is not defined, only <CRLF> will be returned. The reply is followed by **I/OK**.

AT+iUUSR=? Returns the message 'String'. The reply is followed by **I/OK**.

+iUPWD - Remote Firmware Update FTP User Password

Syntax: AT+iUPWD=<password>

Sets user password to logon to the FTP server defined in the AT+iUSRV parameter.

Parameters: <password> = User password to logon to the FTP server. If special characters are used, the password should be specified within quotes. Servers that allow anonymous login usually request an Email address as a password.

Command Options:

AT+iUPWD=" Empty. No user password defined.

Default: Empty. No user password defined.

Result Code:

I/OK If *password* is an empty or legal user password.

I/ERROR Otherwise.

AT+iUPWD~<password> Temporarily set the user password to *password*. The permanent value will be restored after completing the next session.

AT+iUPWD?	Returns a string of asterisk (*) characters indicating the number of characters in the password. If a password is not defined, only <CRLF> will be returned. The reply is followed by I/OK .
AT+iUPWD=?	Returns the message 'String'. The reply is followed by I/OK .

Remote Parameter Update

Syntax:	AT+iRPG= <i>GroupPass</i>	Sets the remote parameter update group/password. Also used to authenticate a remote technician connecting for remote debug purposes.
Parameters:	<i>GroupPass</i> = Group/Password to be used for authentication when accepting W24 parameter updates from a remote web browser.	
Command Options:		
<i>GroupPass</i> = "		Empty: Remote Email Parameter Update and remote Web parameter updates are effectively disabled.
<i>GroupPass</i> = <grp-pass>		grp-pass will be used to authenticate the RPF file retrieved and restrict W24 parameter updates via a remote Web browser.
<i>GroupPass</i> = ""		A password will not be used to authenticate the RPF file retrieved or parameter updates via the Web. Effectively unrestricting any remote W24 parameter updates.
Default:		Empty. No Group/Password defined. When retrieving Email Parameter Update mails shall be skipped. W24 parameter updates via a remote browser are restricted (see note below).
Result Code:		
I/OK		If <i>Group-pass</i> is an empty or legal Group/Password.
I/ERROR		Otherwise.
AT+iRPG~ <i>GroupPass</i>		Temporarily sets the Parameter Update Group/Password to <i>GroupPass</i> . The permanent Group/Password will be restored after completing the next session, whether the session was successful or not.
AT+iRPG?		Report the current Group/Password. If a Group/Password does not exist only <CRLF> will be returned. The reply is followed by I/OK .
AT+iRPG=?		Returns the message 'String'. The reply is followed by I/OK .

Note: This default value is shipped from the factory. The AT+iFD command does not restore RPG to this value.

Secure Socket Protocol Parameters

+iCS - Define the SSL3/TLS Cipher Suite

Syntax: AT+iCS=*n*

Sets the cipher suite to be used in SSL3/TLS negotiations with a secure server.

The default value '0' is the all-cipher selection. With this value, W24 sends its full list of supported ciphers to the server. The server selects the most appropriate cipher to use during the handshake procedure. When a specific value is specified, W24 requires the server to use that specific cipher.

Parameters: *n* = A supported cipher suite code, as defined in RFC2246.

Command Options:

- n*=0 Set cipher suite to 'propose all'. When CS is set to 'propose all', W24 offers all supported cipher suites for SSL3/TLS negotiations. The server selects the most appropriate cipher suite during the handshake procedure.
- n*=4 Set cipher suite to SSL_RSA_WITH_RC4_128_MD5.
- n*=5 Set cipher suite to SSL_RSA_WITH_RC4_128_SHA.
- n*=10 Set cipher suite to SSL_RSA_WITH_3DES_EDE_CBC_SHA.
- n*=47 Set cipher suite to TLS_RSA_WITH_AES_128_CBC_SHA.
- n*=53 Set cipher suite to TLS_RSA_WITH_AES_256_CBC_SHA.
- +1000 Set cipher suite to TLS_RSA_WITH_AES_256_CBC_SHA.

Default: 0 (Propose All).

Result Code:

- I/OK** If *n* is a supported cipher suite code.
- I/ERROR** Otherwise.

AT+iCS? Returns the current cipher suite value. The reply is followed by **I/OK**.

AT+iCS=? Returns the message "0,4,5,10,47,53". The reply is followed by **I/OK**.

+iCA - Define SSL3/TLS Certificate Authority

Syntax: AT+iCA=*tca*

Sets the certificate of the trusted certificate authority. This authority is the one eligible to sign a server's certificate. W24 accepts a server's identity only if its certificate is signed by this authority.

Parameters: *tca* = PEM format DER-encoded X509 certificate.

Command Options:

tca =<CR><CR> Empty: No trusted certificate authority.

tca =<*cert*> *cert* is referenced as the trusted certificate authority's certificate during SSL3/TLS1 socket connection establishment (handshake). W24 establishes an SSL3/TLS1 socket connection only to servers having a certificate authenticated by this certificate authority. W24 expects *cert* to be multiple lines separated by <CR>, beginning with:
-----BEGIN CERTIFICATE-----
and terminating with:
-----END CERTIFICATE-----
Maximum size of *cert* is 1300 characters.

Default: Empty. No trusted Certificate Authority defined.

Result Code:

I/OK If *tca* is an empty or legal certificate.

I/ERROR Otherwise.

AT+iCA? Report the current trusted certificate contents. The reported value displays the Certificate Authority name, certificate validity date range, and the entire PEM contents. If the trusted certificate is empty, only <CRLF> is returned. The reply is followed by **I/OK**.

AT+iCA=? Returns the message "String". The reply is followed by **I/OK**.
Sample PEM format DER-encoded X509 certificate:

```
-----BEGIN CERTIFICATE-----
MIICPDCCAaUCEHC65B0Q2Sk0tjjKewPMur8wDQYJKoZIhvcNAQ
ECBQAwXzELMAkGA1UEBhMCVVMxZzAVBgNVBAoTD1ZlcmlTaWdu
LCBjbmMuMTcwNQYDVQQLZy5DbGFzcyAzIFB1YmVycyBQcm1tYX
J5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MB4XDk2MDEyOTAw
MDAwMFoXDTE4MDgwMTIzNTk1OVowXzELMAkGA1UEBhMCVVMxZz
AVBgNVBAoTD1ZlcmlTaWduLCBjbmMuMTcwNQYDVQQLZy5DbGFz
cyAzIFB1YmVycyBQcm1tYXJ5IENlcnRpZmljYXRpb24gQXV0aG
9yaXR5MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBQqDjXfMe
8huKARS0EN8EQNvjV69qRUCPhAwL0TPZ2RHP7gJYHYX3KqhEBa
rsAx94f56TuZoAqiN91qyFomNFx3InzPRMxnVx0jnvT0Lwdd8K
kMaOIG+YD/isI19wKTakyYbnsZogy10lhec9vn2a/iRFM9x2Fe
0PonFkTGUgWhFpwIDAQABMA0GCSqGSIb3DQEBAQUAA4GBALTm
EivPLCYATxQT3ab7/AoRhIzzKBxni98tsX63/Dolbwdj2wsqF
HMc9ikwFPwTtYmwHYBV4GSXiHx0bH/59AhWM1pF+NEHJwZRDmJ
XNycAA9WjQKZ7aKQRUzkuxCkPfAyAw7xzvjjoyVGM5mKf5p/Afb
dynMk2OmufTqj/ZA1k
-----END CERTIFICATE-----
```

+iCERT - Define SSL3/TLS1 Certificate

Syntax: `AT+iCERT=ct`

Set W24's SSL3/TLS1 certificate.

Some SSL3/TLS1 servers require the client side to authenticate its identity by requesting the client to provide a certificate during the SSL socket negotiation phase. This is called "client side authentication". If the CERT parameter contains a certificate, W24 provides it to the server upon request. W24 also needs a private key (see PKEY parameter) in order to encrypt its certificate before sending it to the server. In addition, the certificate should be signed by a certificate authority accepted by the server for the client side authentication to succeed.

Parameters: `ct` = PEM format DER-encoded X509 Certificate.

Command Options:

`ct =<CR><CR>` Empty: No trusted certificate authority.

`ct =<cert>` `cert` is used as W24's certificate during client side authentication. The certificate must be signed by a certificate authority acceptable by the server.

W24 expects `cert` to be multiple lines separated by `<CR>`, beginning with:

-----BEGIN CERTIFICATE-----

and terminating with:

-----END CERTIFICATE-----

Default: Empty. No trusted Certificate Authority defined.

Result Code:

I/OK If `ct` is an empty or legal certificate.

I/ERROR Otherwise.

`AT+iCERT?` Displays current certificate contents. If the trusted certificate is empty, only `<CRLF>` is returned. The reply is followed by **I/OK**.

`AT+iCERT=?` Returns the message "String". The reply is followed by **I/OK**.

+iPKEY - Define W24's Private Key

Syntax: `AT+iPKEY=pk`

Set W24's private key.

The private key is required to perform an RSA encryption of its certificate (see CERT parameter) when performing client side authentication. Special care should be taken to protect private key contents from unauthorized parties. For this reason, once the private key is stored on W24, it cannot be read - only erased or overwritten.

Parameters: *pky* = PEM format.

Command Options:

pky =<CR><CR> Empty. Any existing private key is erased.

pky =<*pkey*> *pkey* is used as W24's private key to RSA encrypt its certificate during client side authentication.
W24 expects *pkey* to be multiple lines separated by <CR>, beginning with:
-----BEGIN RSA PRIVATE KEY-----
and terminating with:
-----END RSA PRIVATE KEY-----

Default: Empty. No private key defined.

Result Code:

I/OK If *pky* is an empty or legal private key.

I/ERROR Otherwise.

AT+iPKEY? Reports the current private key's strength (number of bits in key). If the key is empty, only <CRLF> is returned. There is no way to retrieve *pkey* contents. The reply is followed by **I/OK**.

AT+iPKEY=? Returns the message "String". The reply is followed by **I/OK**.
Example:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCoMGVcZ3HNFB/cRfWP7vdZrRK+YB+1ez07mA
N6Zcd4C19Xi6M6dmewb6qQ6TRYC1gBhJ+KtMopGoqQ3v1VSu0V
e/ZrjWNxLN9UAtRMubtkGz2j6OCtlx4WsFUWebF8QEE9+3coM
nRqtAdluYEU2F2PTeWUsQfjRQqMBjus/y0wwIDAQABAoGBAKWa
KWOHk1zbENfhpn1XTQNmt4tVuDNHGi6gaERNbM79W54mpsy8oz
HtcWOHy3tZiAjOngyEIH3CXWdxuL0PrkmdSk39+V0EiUA0sRxy
UTb3/LlDU9DpxlYXBYK5Kclq2qH5GBv28QJChG6/dfvu08a1Jy
PwD61iOvBvBye/C7QRAkEA1uU7pT8ejcxzfZLwaBwUift9Y1kpz
rdHYnqJggrhGeZq4bIb8ioOFegB+JKXSxaQZgxUsIkDVzkO/+J
/H8KZKyJBAMhcGEftwPqtZMwyqis7rSUpsewaxg79QYDZVSRw
i5ynLqtqui4dGVsfTbXvtZHRs8uyp3plTFUVFnvPRsUJpukCQE
ZyJzdola+OS8d0EooyMLhWp1y4U2ur2wNF37V6iz/aBJMvPSJ7
MuhP2QpSgeHghax/CFTCRFS1yPzMBFNTcDkCQEhqqo5veNK/4u
xruDJbAr68Ne3gbRKXXUp/tDQ0NqpGEkOQ7EmphyDhHk4J2+1q
XUWBtDm/Q9qmAmyfJ8BBSakCQAaO10MGdUnyFuanp19jRfLB29
oOqMQqyV90r25AxOcNHD8Jsmn5vBYm4wdtR8x84Gh7128RfuBS
8J0hFb90yRY=
-----END RSA PRIVATE KEY-----
```

DHCP Server Parameters

+iDPSZ - DHCP Server Pool Size

Syntax: AT+iDPSZ=<*range*>

Sets number of addresses to be allocated in the IP pool of W24's DHCP server.

Parameters: *range* = number of IP addresses in pool.

Command Options:

range=0-255 When *range*=0 the pool is empty and the DHCP server is inactive. When *range* is set to any number between 1 and 255, and the DIP parameter is defined - the DHCP server becomes active.

Default: 0 - DHCP server is inactive.

Result Code:

I/OK If *range* is an integer between 0 and 255.

I/ERROR Otherwise.

AT+iDPSZ? Reports the current *range* value. The reply is followed by **I/OK**.

AT+iDPSZ=? Returns the message "0-255". The reply is followed by **I/OK**.
Example:

+iDSLT - DHCP Server Lease Time

Syntax: AT+iDSLT=<*time*>

Defines lease time, in minutes, to be granted by W24's DHCP server when assigning IP addresses to clients.

Parameters: *time* = lease time in minutes.

Command Options:

time=0-65535 When *time*=0 lease time is indefinite. Any other value sets a limit on the lease time.

Default: 0 - Indefinite lease time.

Result Code:

I/OK If *time* is an integer between 0 and 65535.

I/ERROR Otherwise.

AT+iDSLT? Reports the current *time* value. The reply is followed by **I/OK**.

AT+iDSLT=? Returns the message "0-65535". The reply is followed by **I/OK**

iRouter Parameters

+iARS - Automatic Router Start

Syntax: `AT+iARS=n`

Causes W24 to automatically enter iRouter mode upon power-up or soft reset.

Upon entering iRouter mode, W24 immediately goes online on the dialup/cellular side. Packets are not buffered during dialup/cellular connection establishment. After establishing the connection, W24 starts the routing service.

Parameters:

`n=0` Do not start iRouter mode upon power-up or soft reset.

`n=1` Enter iRouter mode upon power-up or soft reset.

Default: 0

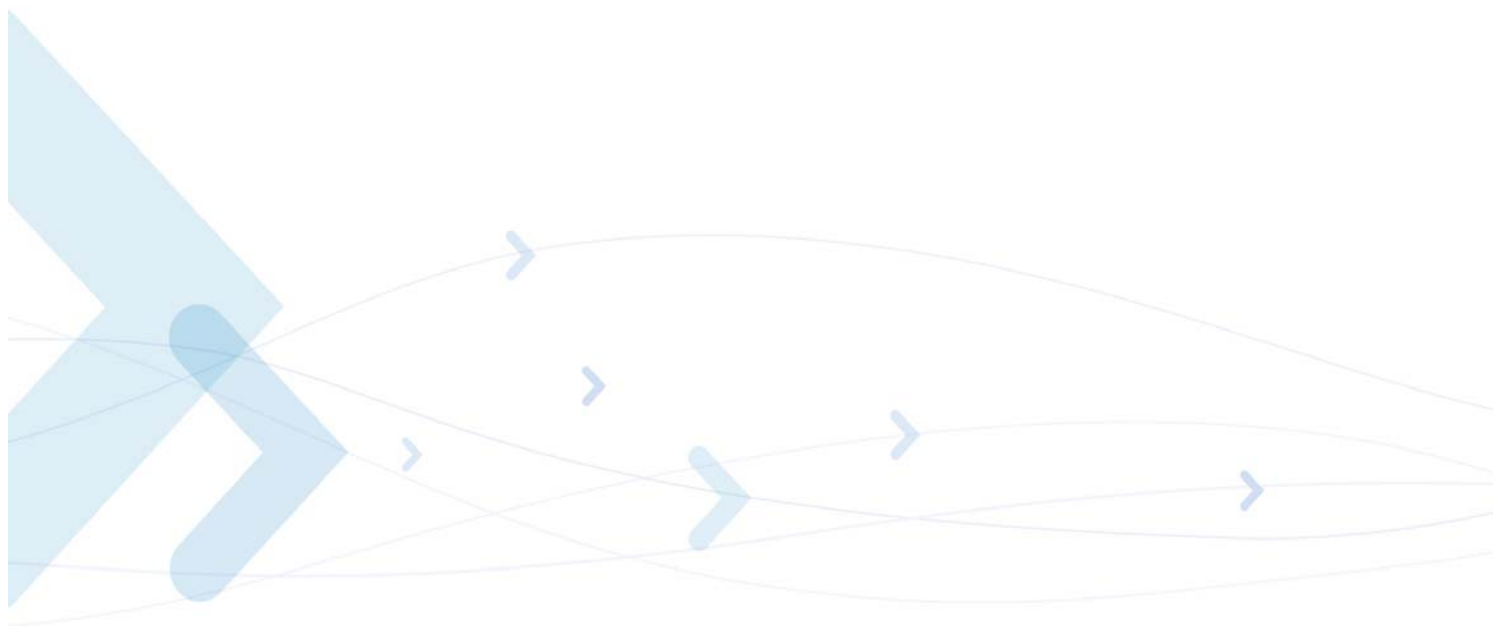
Result Code:

I/OK If *n* is within limits.

I/ERROR Otherwise.

`AT+iDSL?` Reports the current value. The reply is followed by **I/OK**.

`AT+iDSL=?` Returns the message "0, 1". The reply is followed by **I/OK**.



Appendix A: MIME Content Types and Subtypes

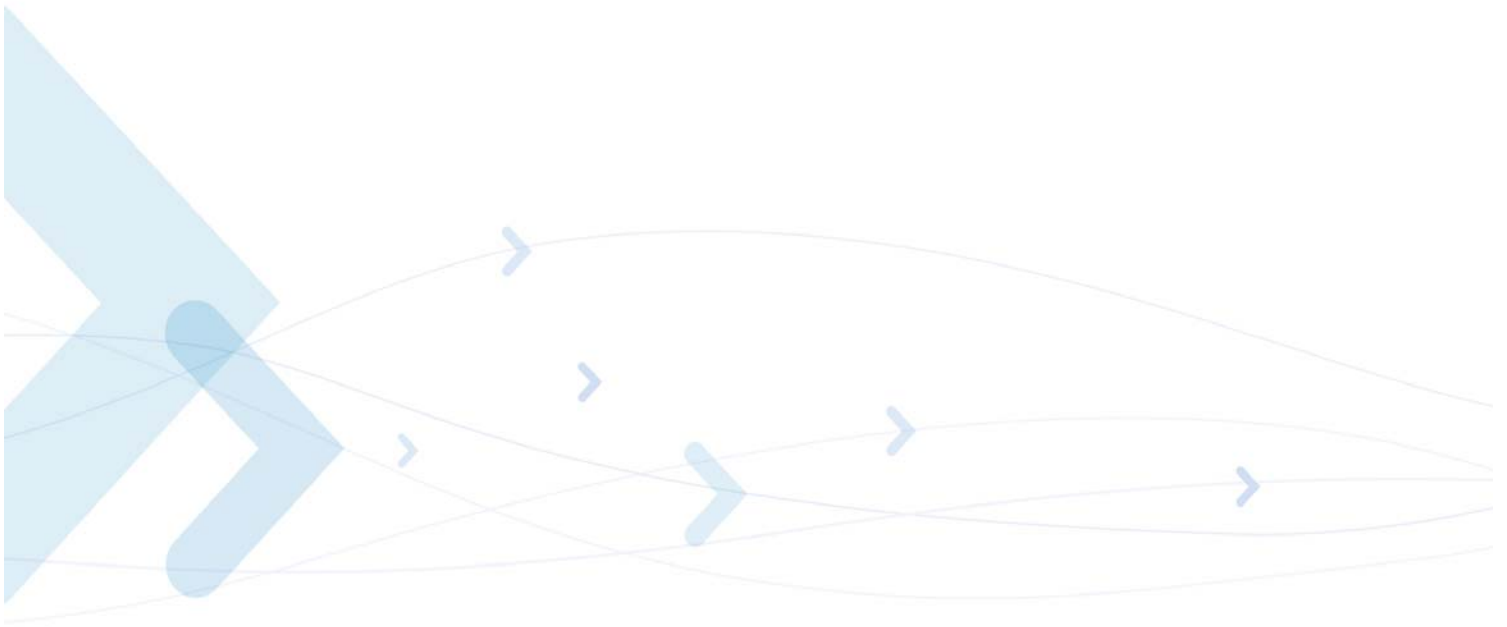
The following tables give MIME content types and subtypes.

Type	Subtype
text	plain
	richtext
	enriched
	tab-seperated-values
	html
	sgml
	vnd.latex-z
	vnd.fmi.flexstor
multipart	Mixed
	Alternative
	Digest
	Parallel
	Appledouble
	header-set
	Form-data
	Related
	Report
	voice-message
	Signed
	Encrypted
message	rfc822
	Partial
	external-body
	News
	http

Type	Subtype	Subtype
application	octet-stream	vnd.music-niff
	Postscript	vnd.ms-artgalry
	Oda	vnd.truedoc
	Atomicmail	vnd.koan
	andrew-inset	vnd.street-stream
	Slate	vnd.fdf
	Wita	set-payment-initiation
	dec-dx	set_payment
	dca-rft	set-registration-initiation
	Activemessage	set-registration
	Rtf	vnd.seemail
	Applefile	vnd.businessobjects
	mac-binhex40	vnd.meridian-slideshow
	news-message-id	vnd.xara
	news-transmission	sgml-open-catalog
	wordperfect5.1	vnd.rapid
	Pdf	vnd.enliven
	Zip	vnd.japannet-registration-wakeup
	Macwriteii	vnd.japannet-verification-wakeup
	Msword	vnd.japannet-payment-wakeup
	remote-printing	vnd.japannet-directory-service
	Mathematica	vnd.intertrust.digibox
	Cybercash	vnd.intertrust.nncp
	commonground	vnd.ms-tnef
	Iges	vnd.svd
	Riscos	
	Eshop	
	x400-bp	
	Sgml	
	cals-1840	
	pgp-encrypted	
	pgp-signature	
	pgp-keys	

Type	Subtype	Subtype
	vnd.framemaker	
	vnd.mif	
	vnd.ms-excel	
	vnd.ms-powerpoint	
	vnd.ms-project	
	vnd.ms-works	

Type	Subtype
image	Jpeg
	Gif
	Ief
	g3fax
	Tiff
	Cgm
	Naplps
	vnd.dwg
	vnd.svf
	vnd.dxf
	Png
	vnd.fpx
	vnd.net-fpx
audio	Basic
	32kadpcm
	vnd.qcelp
video	Mpeg
	Quicktime
	vnd.vivo
	vnd.motorola.video
	vnd.motorola.videop



Appendix B: Sample Parameter Update File

RP_GROUP="111" RP_DEST="00010001"
RP_START_FROM_FACTORY_DEFAULTS=YES

MODEM PARAMETERS:

MIS="ATX4E1&C1&D2M2L2"
XRC=1
BDRM=8

CONNECTION PARAMETERS:

ISP1="7777555"
ISP2="036666555"
USRN="name"
PWD="pass"
DNS1=192.115.106.10
DNS2=192.115.106.11
ATH=1
SMTP="smtp.com"
EMA="name@domain"

POP3 PARAMETERS:

MBX="pop_name"
MPWD="pop_pass"
POP3="pop3.com"
LVS=0
FLS="mymail"

EMAIL STRUCTURE_PARAMETERS:

TOA="email@address.com"

CC1= "cc1@address.com"

CC2= "cc2@address.com"

CC3= "cc3@address.com"

CC4= "cc4@address.com"

SBJ="MySubject"

TOA="someone@hisServer.com"

TO="name"

FRM="me"

REA="myEmail@myServer.com"

BDY="This is my Email"

FN="myfile.txt"

MT=0

MST="text-plain"

CONNECTION TIMEOUT/RETRIES PARAMETERS:

RDL=2

RTO=180

WTC=100

OTHER PARAMETERS:

HDL=5

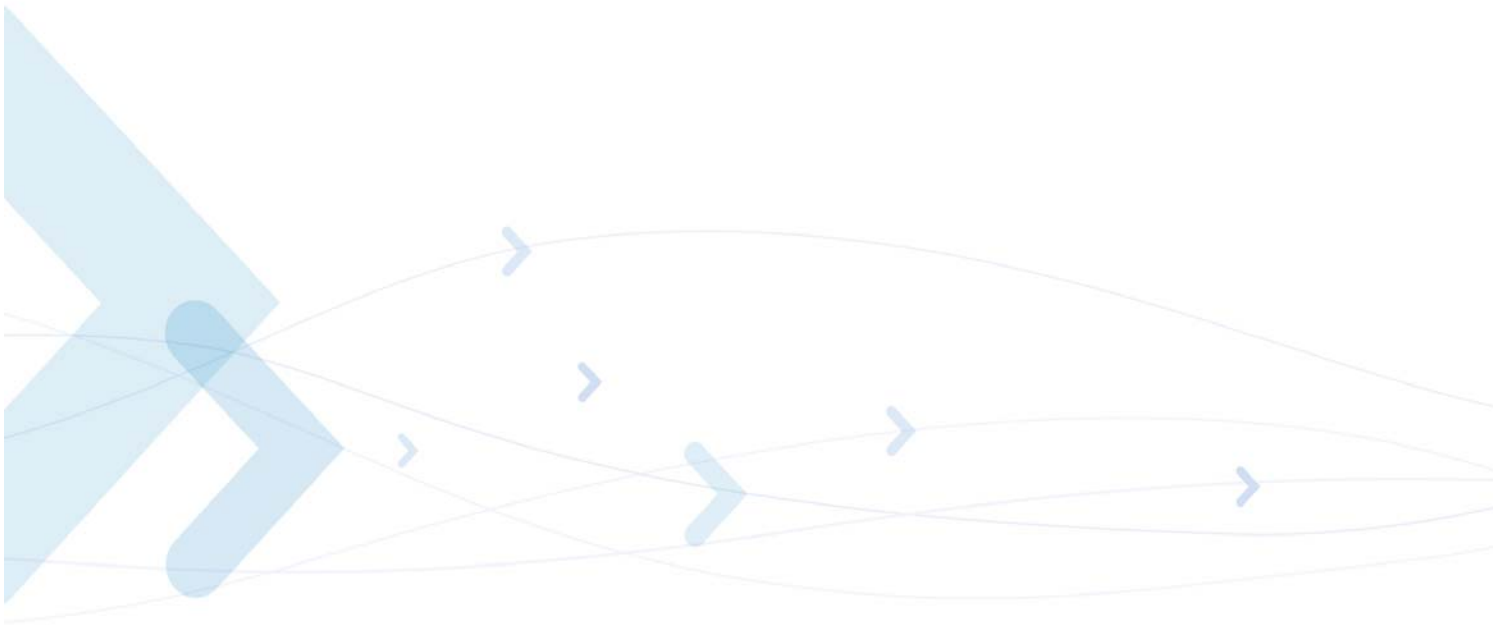
URL="http://www.motorola.com/"

Appendix C: NIST Time Servers

Table C-1: List of NIST Time Servers

Server	IP	Address Location
nist1.aol-ca.truetime.com	207.200.81.113	TrueTime, AOL facility, Sunnyvale, California
nist1.aol-va.truetime.com	205.188.185.33	TrueTime, AOL facility, Virginia
nist1.datum.com	66.243.43.21	Datum, San Jose, California
nist1.datum.com	209.0.72.7	Datum, San Jose, California
nist1.dc.certifiedtime.com	216.200.93.8	Abovnet, Virginia
nist1.nyc.certifiedtime.com	208.184.49.9	Abovnet, New York City
nist1.sjc.certifiedtime.com	208.185.146.41	Abovnet, San Jose, California
nist1-dc.glassey.com	216.200.93.8	Abovenet, Virginia
nist1-ny.glassey.com	208.184.49.9	Abovenet, New York City
nist1-sj.glassey.com	207.126.98.204	Abovenet, San Jose, California
time.nist.gov	192.43.244.18	NCAR, Boulder, Colorado
time-a.nist.gov	129.6.15.28	NIST, Gaithersburg, Maryland
time-a.timefreq.bldrdoc.gov	132.163.4.101	NIST, Boulder, Colorado
time-b.nist.gov	129.6.15.29	NIST, Gaithersburg, Maryland
time-b.timefreq.bldrdoc.gov	132.163.4.102	NIST, Boulder, Colorado
time-c.timefreq.bldrdoc.gov	132.163.4.103	NIST, Boulder, Colorado
time-nw.nist.gov	131.107.1.10	Microsoft, Redmond, Washington
utcnist.colorado.edu	128.138.140.44	University of Colorado, Boulder

Note: Check <http://tf.nist.gov/service/time-servers.html> for updates.



Appendix D: Use Cases

Use Case - Host Mode

W24 supports a Host Mode functionality, when it is connected to an external host (terminal/controller) (see [Figure D-1](#)).

In this mode, the W24 supports two physical connections to a host for two different networks, in spite of the W24 doesn't support several network connections simultaneously. A host uses dynamic switching between the two supported in the W24 networks: WiFi and Cellular (GSM/GPRS). The switching will be done via AT+I commands interface.

For example: when the host lost a connection (coverage problem etc.), it is able to switch itself dynamically to another network for Internet connection. It is a simple variant of seamless network connectivity.

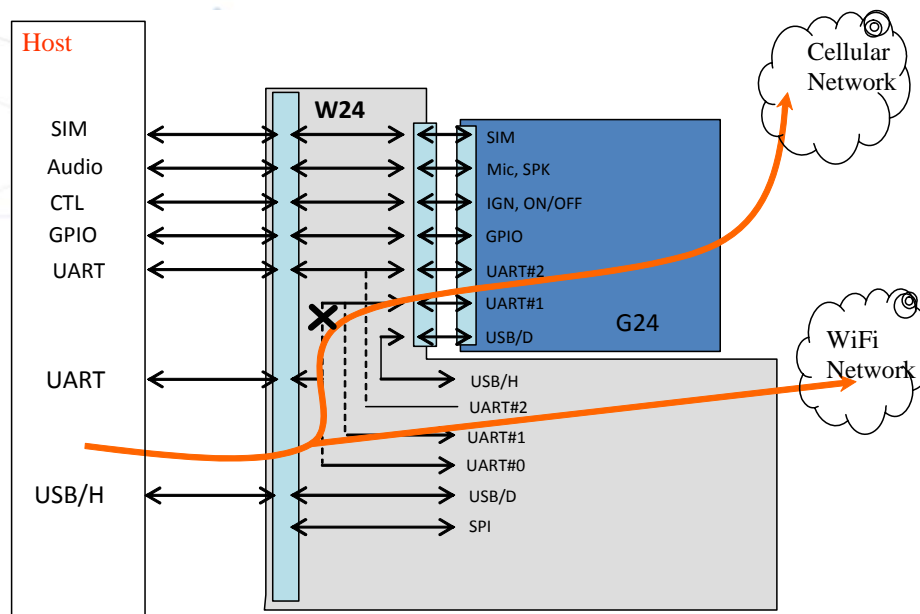


Figure D-1: External Terminal/Host Data Mode

The mode configuration

In order to use the WLAN platform W24 must be notified to switch communication platforms:

Command	Response	Comment
AT+iCPF=1	I/OK I/ONLINE	The W24 is switched to the WLAN communication platform.

The W24 WLAN station associates with an Access Point (AP) according to the following rules:

1. Scan and locate all AP's with a specific SSID
If an SSID is not specified, scan and locate ALL available AP's.
2. Connect to the strongest signal AP from the AP's above.

When the W24 is connected via an AP it will attempt to acquire an IP address from a DHCP server, if its DIP (Default IP) is set to 0.0.0.0. If a DHCP is located and used to acquire the IP address, the Sub-net, Gateway and DNS servers shall also be assigned.

When the DIP is set to an IP address, W24 will use that address and not try to locate a DHCP server. In this case the Sub-net, Gateway and DNS serves must also be assigned manually.

In general, configuring the W24' WLAN communication platform, requires the following settings:

Command	Response	Comment
AT+iWLSI=<SSID string>	I/OK	Setup the required SSID string to use when locating an AP. If this parameter is left blank, W24 will connect to the strongest signal AP.
AT+iDIP=0.0.0.0	I/OK	Configure the W24 to use a DHCP server.
AT+iDOWN	I/OK I/ONLINE	Recycle the connectivity for new parameters to take effect.
AT+iIPA?	172.3.86.192 I/OK	Verify that W24 has connected to AP and received a valid IP address from the DHCP server.

If the AP is connected to a Corporate LAN that has a Gateway to the public Internet, AT+i commands, such as AT+iRLNK:"http://www.google.com/" may be activated in exactly the same way as shown when using the cellular communication platform. The results shall be the same.

Additional advance configuration parameters allow the W24 to be configured for WEP or WPA1/WPA2 security on the WLAN.

In order to use the GSM/GPRS platform W24 must be notified to switch communication platforms:

Command	Response	Comment
AT+iCPF=0	I/OK	The W24 is switched to the GSM/GPRS communication platform.

Command	Response	Comment
AT+iMTYP=2	I/OK	Set the Modem Type to 2 (GPRS).
AT+iXRC=0	I/OK	Set "Blind Dialing" mode, since GPRS modems do not respond with a dial tone.
AT+iISP1=*99#	I/OK	Set the GPRS IP connection dial string.
AT+iMIS="AT+cgdcont=1, \"IP\", \"INTERNET\""	I/OK	Define the Modem Initialization string. This is sent to the G24 before starting an Internet session. It sets the correct APN.

This minimal configuration shall be performed only once. Since all W24 parameters are non-volatile, these settings shall remain intact indefinitely or until a "Factory Defaults" AT+i command is issued.

Once these settings have been configured the W24 can go online through the G24 GPRS connection. This can be verified then by following this command sequence:

Command	Response	Comment
AT+iIPA?	0.0.0.0 I/OK	When the W24 is not yet online its IP is 0.0.0.0.
or AT+iUP	I/OK I/ONLINE	This command W24 to go online using the selected communication platform. The I/ONLINE response signals that W24 is on the Internet.
or AT+iSTRR	I/OK I/ONLINE	This command W24 to go online using the Gateway/Router mode for switching communication platforms seamlessly. I/ONLINE response signals that W24 is on the Internet.
AT+iIPA?	10.170.5.34 I/OK	Now that the W24 is online, it has an assigned IP address.

Note: In case of Gateway/Router mode using, the following parameters and commands should be used to configure the mode behavior:

- Automatic Router Start (AT+iARS) parameter - When set to 1, causes W24 to go online in Gateway/Router mode upon power-up. A default value is 0;
- Start Router (AT+iSTRR) command - Causes W24 to go online in Gateway/Router mode.
- Stop Router (AT+iSTPR) command - Causes W24 to exit Gateway/Router mode and go offline.
- Inactivity Timeout (AT+iIATO) parameter - When in Gateway/Router mode, if no activity is detected for the period of time specified by this parameter, W24 disconnects its modem side and go offline.

Use Case - Gateway/Router Mode

W24 together with GSM/GPRS modem (G24) supports functionality as a gateway/router between 802.11 (WLAN) and Cellular networks (see [Figure D-2](#)).

For this mode the W24 shall be configured as a Gateway/Router (see an example of the configuration command sequence in the table below). In this mode WLAN stations communicating to W24 on the WLAN can use the W24 as a gateway to the Internet through the Cellular modem. The W24 will handle routing and Network Address Translation between the Cellular uplink to the Internet and local WLAN stations. In this mode W24 coupled with G24 can be used without any host connected, as a standalone device.

Using the CPF parameter, you can select either one of the communication platforms at a time. When CPF=0, AT+i commands affect the dialup/cellular side; when CPF=1, they are directed at the WiFi side. While accepting AT+i commands, W24 continues to route packets seamlessly between the two platforms. W24's responses to AT+i commands depend on the CPF value, as well. For example, the IP returned by AT+iIPA? command while CPF=1 is the WLAN-side IP.

Upon entering Gateway/Router mode, W24 immediately goes online on the WiFi side and the Cellular. If an IP is not configured, W24 attempts to resolve one using its DHCP client. After establishing both connections, W24 holds two IP addresses, each associated with one of the physical interfaces.

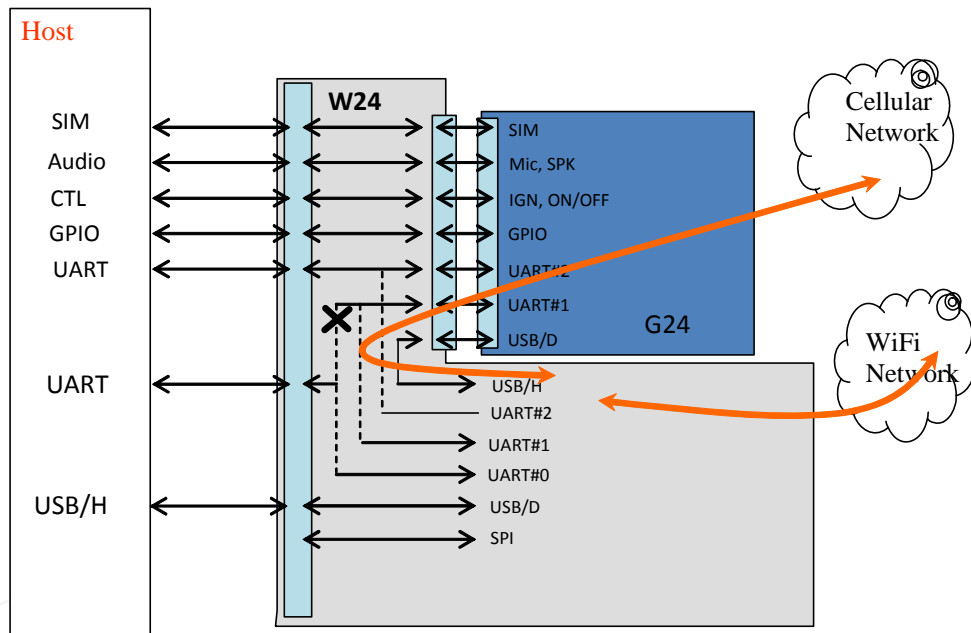


Figure D-2: Gateway/Router Data Mode

For example of gateway/router mode configuration see [“The mode configuration” on page 1](#).

In Gateway/Router mode, W24 can be configured using the following methods:

- AT+i commands coming from the host application.
- From the WiFi end, W24's internal configuration website can be accessed by any browser.

Routing inside of W24

W24 routes packets between the two communication platforms according to the following rules:

- IP packets received on the WLAN end, where the destination address is different than its own, are routed to the Cellular end.
- IP packets received on the Cellular end are normally routed to the WLAN end, with the exception of a fixed port set aside for W24 web-based configuration.

Routing is determined at the IP layer, where source and destination IP addresses are extracted. Before routing a packet between the physical interfaces, W24 exchanges the relevant source or destination IP address, as normally performed by a Network Address Translation (NAT) process, to conceal the WLAN end's 'false' address. The IP exchange is performed according to the following rules:

- In IP packets sent from the WLAN end, the source IP is exchanged with W24's Cellular end IP.
- In IP packets arriving from the Cellular end, the destination IP is exchanged with the host's WLAN end IP, unless received on a fixed port set aside for W24 configuration.

- Port numbers are not manipulated.

In addition, the IP packet's TTL field is decremented and the TCP and IP checksums are re-calculated.

Use Case - Integrated Host (Java) Mode

W24 together with G24 supports functionality as Integrated Host (Java) (see [Figure D-3](#)).

In this mode Java (inside of the G24) takes role of a Host. Via the USB or UART1 port of G24, Java (MIDlet) is able to send to W24 AT+I commands, meaning to configure and control it as for example a Host/Terminal. More then this, Java is able to make a Cellular (GSM/GPRS) connection and control a data traffic from/to the connection. So, the G24's Java together with W24 capabilities will be able to operate as a gateway between the two supported by W24 networks.

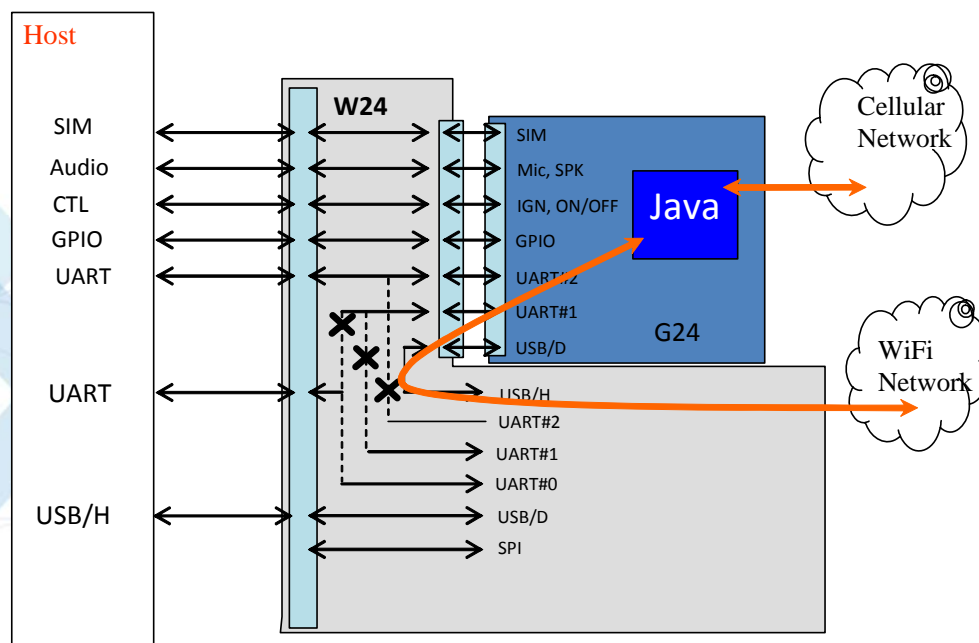


Figure D-3: Integrated Host (Java) Mode

In this mode W24 coupled with G24 can be used without any host connected, as a standalone device.

Index

A

AT+i Commands Overview, 1-1

AT+i Commands Reference

Ad-Hoc Networks, 2-55

Connection

+iBDRA, Force W24 into Auto Baud Rate Mode, 2-6

+iDOWN, Terminate Internet Session, 2-8

+iPING, Send a PING Request to a Remote Server, 2-9

+iTUP, Triggered Internet Session Initiation, 2-7

+iUP, Initiate Internet Session, 2-6

DHCP Client, 2-49

DHCP Server, 2-50

Direct Socket Interface

+iGPNM, Get Peer Name for a Specified Socket, 2-38

+iLSST, Get a Listening Socket's Active Connection Status, 2-35

+iLTCP, Open a TCP Listening Socket, 2-34

+iSCLS, Close Socket, 2-40

+iSCS, Get a Socket Connection Status Report, 2-36

+iSDMP, Dump Socket Buffer, 2-39

+iSFSH[%], Flush Socket's Outbound Data, 2-39

+iSRCV, Receive a Byte Stream from a Socket's Input Buffer, 2-38

+iSSND[%], Send a Byte Stream to a Socket, 2-37

+iSST, Get a Single Socket Status Report, 2-35

+iSTCP, Open and Connect a TCP Socket, 2-33

+iSUDP, Open a Connectionless UDP Socket, 2-33

E-mail Retrieve

+iRMH, Retrieve Mail Header, 2-13

+iRML, Retrieve Mail List, 2-13

+iRMM, Retrieve Mail Message, 2-14

E-mail Send Commands

+iE*, Terminate Binary E-Mail, 2-12

+iEMA, Accept ASCII-Coded Lines for E-Mail Send, 2-10

+iEMB, Accept Binary Data for Immediate E-Mail Send, 2-11

File Transfer Protocol

+i[@]FOPN, FTP Open Session, 2-22

+iFAPN, FTP Open File for Appending, 2-27

+iFCLF, FTP Close File, 2-28

+iFCLS, FTP Close Session, 2-29

+iFCWD, FTP Change Working Directory, 2-24

+iFDEL, FTP Delete File, 2-28

+iFDL, FTP Directory Listing, 2-23

+iFDNL, FTP Directory Names Listing, 2-23

+iFMKD, FTP Make Directory, 2-24

+iFRCV, FTP Receive File, 2-25

+iFSND, FTP Send File Data, 2-27

+iFSTO, FTP Open File for Storage, 2-26

+iFSZ, FTP File Size, 2-25

File Transfer Protocol (FTP) Theory of Operation, 2-92

Flow Control, 2-66

HTTP Client Interface

+iRLNK, Retrieve Link, 2-17

+iSLNK, Submit a POST Request to a Web Server, 2-18

IP Registration, 2-47

iRouter Mode, 2-51

+iSTPR, Stop Router, 2-54

+iSTRR, Start Router, 2-53

MIME Encapsulated E-Mail Messages, 2-62

Network Time Client, 2-61

Nonvolatile Parameter Database, 2-102

+iADCD, ADC Delta, 2-125

+iADCL, ADC Level, 2-125

+iADCP, ADC GPIO Pin, 2-126

+iADCT, ADC Polling Time, 2-126

+iARS, Automatic Router Start, 2-193

+iATH, Set PPP Authentication Method, 2-129

+iAWS, Activate WEB Server Automatically, 2-116

+iBDRD, Baud Rate Divider, 2-115

+iBDRF, Define a Fixed Baud Rate on Host Connection, 2-114

+iBDRM, Define a Fixed Baud Rate on W24<->Modem Connection, 2-115

+iCA, Define SSL3/TLS Certificate Authority, 2-189

+iCCn, Define Alternate Addressee <n>, 2-149

+iCERT, Define SSL3/TLS1 Certificate, 2-190

+iCKSM, Checksum Mode, 2-123

+iCPF, Active Communications Platform, 2-118

+iCS, Define the SSL3/TLS Cipher Suite, 2-188

+iCTT, Define Content Type Field in POST Request, 2-152

+iDELF, Email Delete Filter String, 2-146

+iDF, IP Protocol 'Don't Fragment' Bit Value, 2-122

+iDIP, W24 Default IP Address, 2-156

+iDMD, Modem Dial Mode, 2-108

+iDNSn, Define Domain Name Server IP Address, 2-134

+iDPSZ, DHCP Server Pool Size, 2-192

+iDSLTL, DHCP Server Lease Time, 2-192

+iDSTD, Define Daylight Savings Transition Rule, 2-141

+iDSTR, Define Disconnection String for SerialNET Mode, 2-173

+iFCHR, Flush Character, 2-175

+iFD, Restore All Parameters to Factory Defaults, 2-107

+iFLS, Define Filter String, 2-145

+iFLW, Set Flow Control Mode, 2-118

+iFN, Attachment File Name, 2-151

+iFRM, Email 'From' Description/Name, 2-149

+iGMTO, Define Greenwich Mean Time Offset, 2-141

+iHDL, Limit Number of Header Lines, 2-144

+iHIF, Host Interface, 2-123

+iHSRV | +iHSRn, Host Server Name/IP, 2-172

+iHSS, Assign Special Characters to Hosts, 2-173

- +iHSTN, W24 Network Host Name, [2-171](#)
- +iIATO, Inactivity Timeout, [2-177](#)
- +iIPA, Active IP Address, [2-157](#)
- +iIPG, IP Address of the Gateway, [2-157](#)
- +iISPn, Set ISP Phone Number, [2-129](#)
- +iLATI, TCP/IP Listening Socket to Service Remote AT+i Commands, [2-116](#)
- +iLPRT, SerialNET Device Listening Port, [2-174](#)
- +iLVS, 'Leave on Server' Flag, [2-132](#)
- +iMACA, MAC Address of W24, [2-156](#)
- +iMBTB, Max Bytes To Buffer, [2-174](#)
- +iMBX, Define POP3 Mailbox Name, [2-138](#)
- +iMCBF, Maximum Characters before Socket Flush, [2-176](#)
- +iMIF, Modem Interface, [2-124](#)
- +iMIS, Modem Initialization String, [2-109](#)
- +iMPS, Max PPP Packet Size, [2-113](#)
- +iMPWD, Define POP3 Mailbox Password, [2-139](#)
- +iMRST, Turn the W24 Off, [2-120](#)
- +iMST, Media Subtype String, [2-150](#)
- +iMT, Media Type Value, [2-150](#)
- +iMTTF, Max Timeout to Socket Flush, [2-175](#)
- +iMTYP, Set Type of Modem Connected to W24, [2-110](#)
- +iNTSn, Define Network Time Server, [2-139](#)
- +iPDSn, Define PING Destination Server, [2-142](#)
- +iPFR, PING Destination Server Polling Frequency, [2-142](#)
- +iPGT, PING Timeout, [2-112](#)
- +iPKEY, Define W24's Private Key, [2-190](#)
- +iPOP3, Define POP3 Server Name, [2-137](#)
- +iPSE, Set Power Save Mode, [2-119](#)
- +iPTD, SerialNET Packets to Discard, [2-184](#)
- +iPWD, Define Connection Password, [2-130](#)
- +iRAP, Password for RAS Authentication, [2-155](#)
- +iRAR, RAS RINGs, [2-154](#)
- +iRAU, Define RAS Login User Name, [2-154](#)
- +iRDL, Number of Times to Redial ISP, [2-131](#)
- +iREA, Return Email Address, [2-148](#)
- +iRRA, W24 Readiness Report Activation, [2-127](#)
- +iRRHW, W24 Readiness Hardware Pin, [2-128](#)
- +iRRMA, IP Registration Mail Address, [2-168](#)
- +iRRRL, IP Registration Return Link, [2-170](#)
- +iRRSV, IP Registration Host Server Name, [2-169](#)
- +iRRWS, IP Registration Web Server, [2-170](#)
- +iRTO, Delay Period between Redials to ISP, [2-132](#)
- +iS100, Define Wait Interval Between Wakeup Events, [2-121](#)
- +iS102, Define Delay after Wakeup before Sending Data, [2-120](#)
- +iSBJ, Email Subject Field, [2-146](#)
- +iSDM, Service Disabling Mode, [2-121](#)
- +iSDT, SerialNET Dialup Timeout, [2-182](#)
- +iSMA, SMTP Authentication Method, [2-136](#)
- +iSMP, Define SMTP Login Password, [2-137](#)
- +iSMTP, Define SMTP Server Name, [2-135](#)
- +iSMU, Define SMTP Login User Name, [2-136](#)
- +iSNET, Subnet Address, [2-158](#)
- +iSNRD, SerialNET Device Re-Initialization Delay, [2-180](#)
- +iSNSI, SerialNET Device Serial Interface, [2-178](#)
- +iSPN, SerialNET Server Phone Number, [2-181](#)
- +iSTYP, SerialNET Device Socket Type, [2-179](#)
- +iSWT, SerialNET Wake-Up Timeout, [2-183](#)
- +iTO, Email 'To' Description/Name, [2-147](#)
- +iTOA, Define Primary Addressee, [2-147](#)
- +iTTO, TCP Timeout, [2-111](#)
- +iTTR, TCP Retransmit Timeout, [2-113](#)
- +iUEN, Remote Firmware Update Flag, [2-184](#)
- +iUFn, User Fields and Macro Substitution, [2-143](#)
- +iUPWD, Remote Firmware Update FTP User Password, [2-186](#)
- +iURL, Default URL Address, [2-152](#)
- +iUSRN, Define Connection User Name, [2-130](#)
- +iUSRV, Remote Firmware Update Server Name, [2-185](#)
- +iUUSR, Remote Firmware Update FTP User Name, [2-186](#)
- +iWKYn, Wireless LAN WEP Key Array, [2-166](#)
- +iWLCH, Wireless LAN Communication Channel, [2-159](#)
- +iWLKI, Wireless LAN Transmission WEP Key Index, [2-160](#)
- +iWLKn, Wireless LAN WEP Key Array, [2-161](#)
- +iWLPP, Personal Shared Key Pass-Phrase, [2-162](#)
- +iWLPS, Wireless LAN Power Save, [2-162](#)
- +iWLSI, Wireless LAN Service Set Identifier, [2-159](#)
- +iWLWM, Wireless LAN WEP Mode, [2-160](#)
- +iWPPn, Pre-Shared Key Passphrase Array, [2-166](#)
- +iWPSI, Periodic WiFi Scan Interval, [2-163](#)
- +iWPWD, Password for Application Website Authentication, [2-153](#)
- +iWROM, Enable Roaming in WiFi, [2-163](#)
- +iWSEC, Wireless LAN WPA Security, [2-168](#)
- +iWSIn, Wireless LAN Service Set Identifier Array, [2-165](#)
- +iWSRH, SNR High Threshold, [2-164](#)
- +iWSRL, SNR Low Threshold, [2-164](#)
- +iWSTn, Wireless LAN Security Type Array, [2-167](#)
- +iWTC, Wait Time Constant, [2-111](#)
- +iXFH, Transfer Headers Flag, [2-144](#)
- +iXRC, Extended Result Code, [2-108](#)
- +NTOD, Define Network Time-of-Day Activation Flag, [2-140](#)
- Remote Parameter Update, [2-187](#)
- Remote AT+i Service, [2-100](#)
- Remote Firmware Update, [2-71](#)
 - +iRFU, Remote Firmware Update, [2-72](#)
- Report Status
 - +i[!]RPI, Report Status, [2-1](#)
 - Status Message Format, [2-2](#)
- Secure Socket Protocol, [2-57](#)
 - +i[@]FOPS, Secure FTP Open Session, [2-59](#)
 - +iSSL, Secure Socket Connection Handshake, [2-59](#)
- Secure Socket Protocol Theory of Operation, [2-95](#)
- SerialNET Mode Initiation
 - +iSNMD, Activate SerialNET Mode, [2-19](#)
- SerialNET Theory of Operation, [2-85](#)
- Special Modem Commands

+iMCM, Issue Intermediate Command to Modem, [2-41](#)
Telnet Client
+iTBSN[%], Telnet Send a Byte Stream, [2-31](#)
+iTCLS, Telnet Close Session, [2-32](#)
+iTFSH[%], Flush Telnet Socket's Outbound Data, [2-32](#)
+iTOPN, Telnet Open Session, [2-30](#)
+iTRCV, Telnet Receive Data, [2-30](#)
+iTSND, Telnet Send Data Line, [2-30](#)
Telnet Client Operation, [2-94](#)
W24 Embedded Web Server, [2-75](#)
W24 Parameter Update, [2-73](#)
W24 RAS Server, [2-82](#)
Web Server Interface
+iWNXT, Retrieve Next Changed Web Parameter, [2-21](#)
+iWWW, Activate Embedded Web Server, [2-21](#)
Wireless LAN Mode
+iWLBW, WLAN B Mode, [2-44](#)
+iWLGW, WLAN G Mode, [2-44](#)
+iWLPW, Set WLAN Tx Power, [2-43](#)
+iWLTR, Wireless LAN Transmission Rate, [2-42](#)
+iWRFU, WLAN Radio Up, [2-43](#)
+iWRST, Reset WLAN Chipset, [2-44](#)

Roaming Mode, [2-44](#)

G

General AT+i Command Format, [1-7](#)

M

MIME Content Types and Subtypes, [A-1](#)

N

NIST Time Servers, [C-1](#)

S

Sample Parameter Update File, [B-1](#)

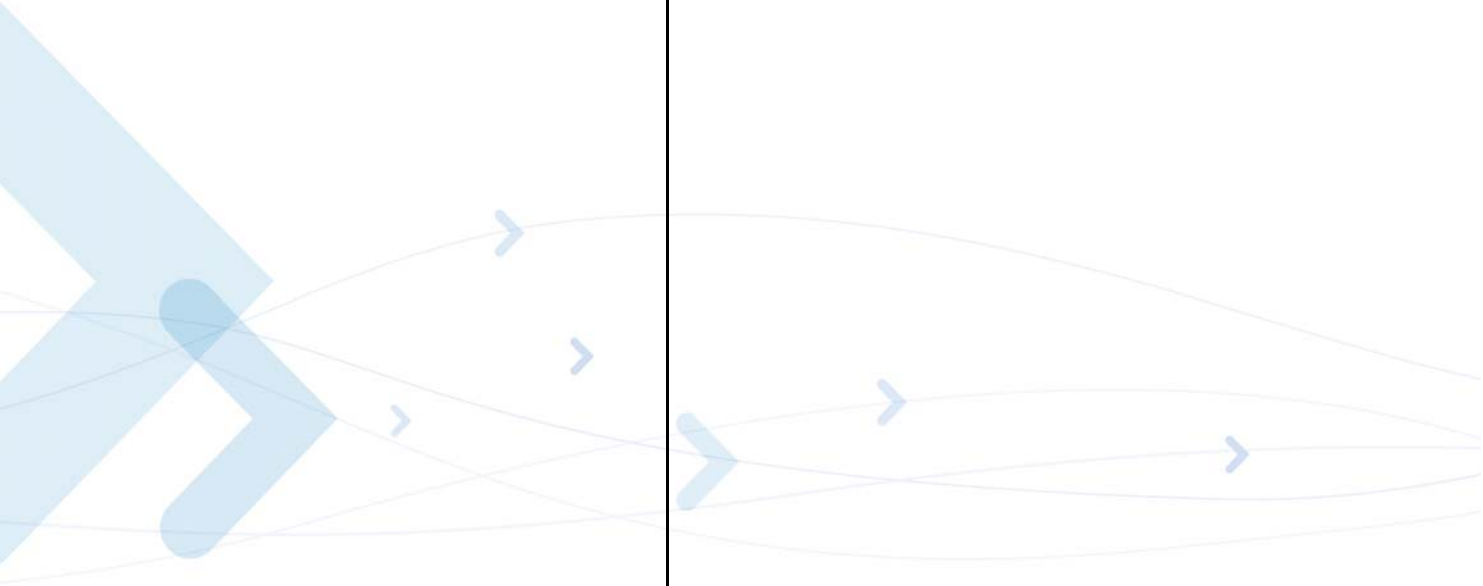
U

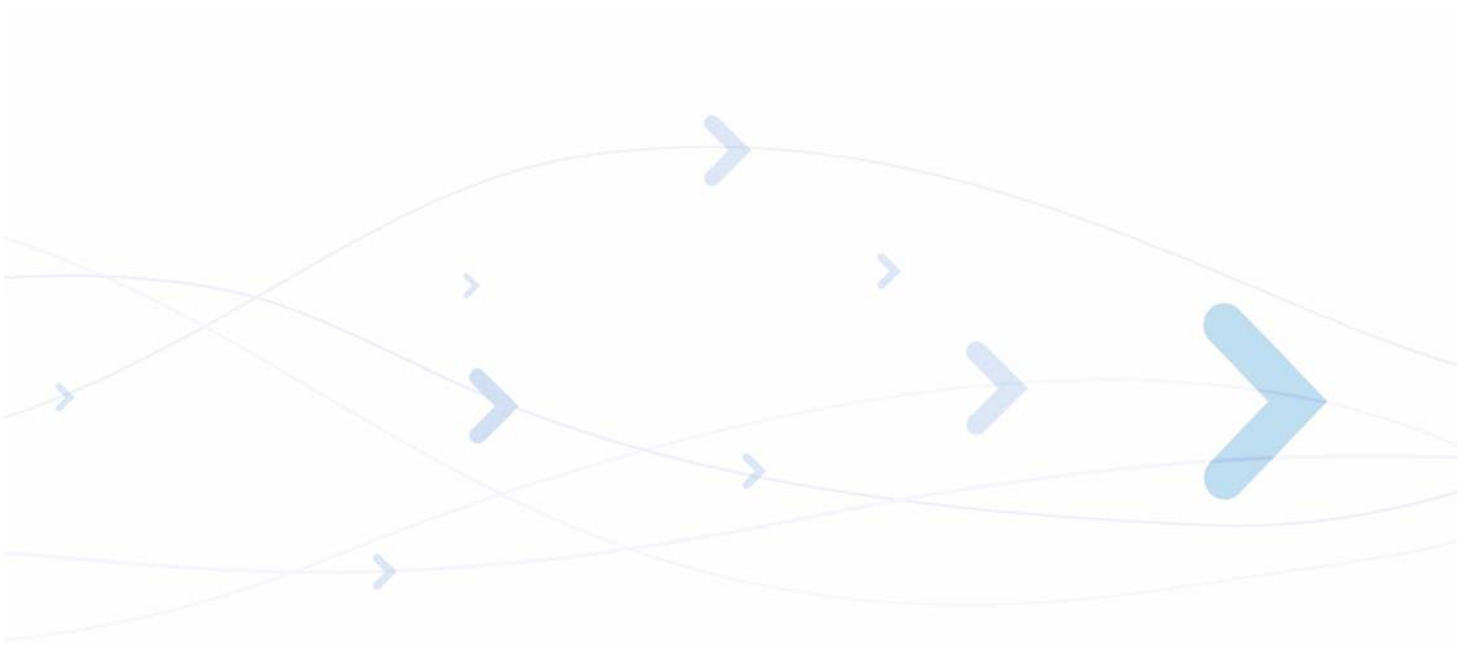
Use Case

Gateway/Router Mode, [D-4](#)

Host Mode, [D-1](#)

Integrated Host (Java) Mode, [D-6](#)





MOTOROLA and the Stylized M Logo are registered in the US Patent & Trademark Office.

All other product or service names are the property of their respective owners.

©Copyright 2007 Motorola, Inc.

Java? Technology and/or J2ME? : Java and all other Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX? : UNIX is a registered trademark of The Open Group in the United States and other countries.



6802985C10-A